Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 5, N. 1, 2017.

Trabalho apresentado no CNMAC, Gramado - RS, 2016.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Engenharia reversa em Jogos de Minoria

Amaury S. Amaral[1]
Departamento de Contabilidade, PUC, São Paulo, SP
Jair Donadelli[2]
Centro de Matemática, Computação e Cognição, UFABC, Santo André, SP
Fernando F. Ferreira[3]
Núcleo Interdisciplinar de Sistemas Complexos, USP, São Paulo, SP

Resumo. Neste trabalho estamos interessados em descobrir as estratégias utilizadas por um conjunto de investidores que são recompensados por comprar quando a maioria esta vendendo e por vender quando a maioria esta comprando (lei da oferta e procura). As decisões de cada jogador são descrita por matrizes de estratégias num Jogo de Minoria. Chamamos de engenharia reversa o conjunto de procedimentos que permitem conhecer estas matrizes a partir do valor agregado das decisões individuais. Para encontrar tais matrizes implementamos dois algoritmos, um com a técnica conhecida por Algoritmos Genéticos e outro baseado em Programação Linear Inteira Mista. Também, a partir das matrizes foi possível selecionar os jogadores mais influentes. Infiltramos jogadores com essas estratégias mais influentes em um novo jogo e observamos resultados positivos em seus desempenhos.

Palavras-chave. Jogos de Minoria, Algoritmos Genéticos, Programação Linear Inteira Mista, Engenharia Reversa.

1 Introdução

O *Jogo de Minoria* foi proposto pelos físicos Yi-Cheng Zhang e Damien Challet em 1997 e, em suas muitas variantes, é um modelo simples que mostra como jogadores (egoístas) cooperam na ausência de comunicação. No jogo há um número ímpar de jogadores os quais têm que escolher, baseados em uma estratégia, uma jogada individual dentre duas opções e de forma independente dos outros jogadores. As estratégias dos jogadores que optam pela jogada da minoria são recompensadas. Esse jogo ganhou muita popularidade como um modelo simples, mas realista, do funcionamento dos mercados financeiros [1, 2].

Sua notoriedade deve-se ao fato de ser um arcabouço teórico para muitas situações nas quais os retornos das ações de agentes num coletivo dependem das decisões dos outros agentes que enfrentam exatamente o mesmo problema. Isso ocorre, por exemplo, quando as empresas precisam escolher se querem entrar em um novo mercado, na possibilidade de investir em uma nova tecnologia, nos comerciantes que decidem quando comprar ou vender uma ação. Em tais situações os agentes têm tanto que coordenar quanto competir.

[1]asamaral@pucsp.br
[2]jair.donadelli@ufabc.edu.br
[3]ferfff@gmail.com

A predição de mercados financeiros é um objeto de estudo tanto dos profissionais de finanças quanto de acadêmicos. Embora tenham certa eficiência, sabe-se que os mercados financeiros são marcadamente imprevisíveis e a inexistência dessa previsibilidade reforça o fator risco que acaba incorporado ao elemento preço. Este fato está relacionado com a hipótese de mercado eficiente. No entanto, na prática, ocorrem momentos de violação desta hipótese e surgem janelas preditivas. A engenharia reversa do Jogo de Minoria seria uma forma de identificar estes momentos em que o mercado se torna preditível [3, 4].

Um comportamento interessante pode surgir quando os jogadores interagem a longo prazo. Nessa hipótese, surgem determinados jogadores que passam a ser predominantes no desempenho global. O jogo, como especificado na próxima seção, é uma coleção de funções cujos valores dão a dinâmica do processo mas, também, nos permite o uso das ferramentas da Análise de funções booleanas que quantificam o comportamento dos agentes. Nossos primeiros estudos indicam que a *influência*, como definida em [5], se apresenta como parâmetro bastante interessante na identificação do grupo do jogadores da minoria.

2 Descrição do modelo

O jogo é composto por agentes N agentes, cada um recebe um conjunto de estratégias *adhoc* no início do jogo, o qual evolui em tempo discreto $t \in \mathbb{N}$. Em cada rodada do jogo o agente i toma uma decisão $a_i(t) \in \{-1, 1\}$ baseado na estratégia com a melhor performance até então e no *estado* do sistema, ou seja, um vetor $\mu(t) \in \{-1, 1\}^m$ que representa as escolhas da maioria nas últimas m rodadas, m é o *tamanho da memória* e é um dos parâmetros do jogo. A estratégia j do agente i é uma função $R_{i,j} : \{-1, 1\}^m \to \{-1, 1\}$ que associa a cada estado uma decisão. Todos os agentes tem o mesmo número S (fixo) de estratégias.

A grandeza medida ao longo da dinâmica é o *excesso de demanda*, para cada instante,

$$A(t) := \sum_{i=1}^{N} a_i(t). \tag{1}$$

O resultado de cada rodada é a função $-\text{sinal}\big(A(t)\big)$. Uma medida H, chamada de *previsibilidade*, é dada pela média temporal de $A(t)$ condicionada a um estado μ. O comportamento de H mede-se em função do parâmetro $\alpha := 2^m/N$ chamado de *parâmetro de controle*.

O grupo de agentes que tomou a decisão que forma a minoria vence a rodada e recompensas são dadas para as estratégias do lado ganhador. Cada agente i mantém um registro de avaliação $U_{i,j}$ dado por

$$U_{i,j}(t+1) := U_{i,j}(t) - A(t) \cdot R_{i,j}\left(\mu(t)\right)) \tag{2}$$

chamado *payoff*, para cada uma de suas estratégias independentemente delas terem sido usadas ou não. No instante t do jogo cada agente i seleciona sua melhor estratégia $s_i^*(t)$

$$s_i^*(t) := \underset{j \in \{1,\ldots,S\}}{\arg\max}\, U_{i,j}(t).$$

Para mais detalhes, vejam os trabalhos [6, 7].

3 A engenharia reversa

O objetivo de uma engenharia reversa é reconstruir as matrizes de estratégias dos jogadores a partir de uma série temporal gerada pela dinâmica do Jogo de Minoria. Propomos a seguir dois algoritmos usando técnicas diferentes de otimização para realizar esta tarefa: um via algoritmo genético e outro via programação linear inteira.

O algoritmo genético, o qual nos referimos por AG, parte de um conjunto de estratégias (cromossomos), opera mutações nos vetores (linhas das matrizes) que vão simulando novos descendentes, tendo como função objetivo minimizar a diferenças entre os excessos de demanda alvo e simulado. No outro algoritmo, o qual nos referimos por PLI0-1, ocorre uma otimização linear para cada linha. Encontramos, os valores ótimos para cada variável (linha da matriz), com o que se obtém ao final uma matriz construída em função da otimização dos valores booleanos com erro controlado.

A qualidade do ajuste entre as matrizes alvo e simuladas será medida, nos algoritmos, pela distância de Hamming entre duas linhas da matrizes de estratégias (alvo e simulada).

3.1 Algoritmo genético

O algoritmo genético busca encontrar as matrizes de entrada iniciais das estratégias dos jogadores que chamaremos de M_1 e M_2. Para que isso ocorra nos valemos dos seguintes passos:

1. para cada rodada t do Jogo de Minoria obtemos dois parâmetros: $A(t)$ e $\mu(t)$. Esses parâmetros são armazenados em dois vetores de tamanho $T_{\max}$ (número total de rodadas).

2. Baseado nesses parâmetros criamos dois vetores de $\{-1, 1\}^N$, correspondentes às linhas da matrizes. Esse processo utiliza os *payoffs*.

   (2.1) Inicia-se assim pelos primeiros parâmetros armazenados $A(1)$ e $\mu(1)$ e cria-se dois vetores. Por exemplo para $A(1) = 3$ de $N = 5$ (jogadores) temos um possível vetor $v_1 = (1, -1, 1, 1, 1)$ e outro possível vetor $v_2 = (1, 1, -1, 1, 1)$.

   (2.2) Usa-se esses dois vetores para iniciar o processo de cruzamento.

3. O algoritmo então simula o jogo cujos elementos obtidos são ajustados através do processo de *crossover*. Procede-se ao ajuste dos vetores v_1 e v_2 tomando-se como referência final a soma desses mesmos elementos, os quais serão comparados a cada momento aos parâmetros $A(t)$ e $A(t + 1)$.

3.2 Algoritmo em programação linear inteira binária

Seguimos o seguinte roteiro para obter as matrizes M_1' e M_2' que estimam as matrizes originais M_1 e M_2. Inicialmente temos os vetores dos excessos de demanda e dos estados de cada rodada.

- Definimos um limitante inferior como o menor valor de A dentre todos os valores de excesso de demanda que ocorrem num dado estado μ, denotado $l^\alpha = \min\{A|\mu\}$ e, analogamente, o limitante superior $l^{\beta} = \max\{A|\mu\}$. Na verdade queremos encontrar os valores dos conjuntos situados entre tais mínimos e máximos.

- Define-se também os *payoffs* sintéticos equivalentes aos mesmos da simulação do jogo. Tomemos V_1 e V_2 vetores de tamanho N inicializados com zeros.
- O procedimento tem início na obtenção do excesso de demanda $A(1)$ no estado $\mu(1)$ e os limitantes inferior l^{α} e superior l^{β} do estado $\mu(1)$.
- Definimos os vetores correspondentes a uma linha da matriz $\omega^1 \in M_1'$ e $\omega^2 \in M_2'$, são encontrados de acordo com a seguinte regra: $R_{t,i} \leftarrow j$ em que j é tal que $V_i^j = \max\{V_i^1, V_i^2\}$; em caso de igualdade, $V_i^1 = V_i^2$, sorteia-se $j \in \{1, 2\}$.
- Executamos a formulação programaLinearPorLinha(R(t), ω^1, ω^2, $A(t)$, $l^{\alpha}_{\mu(t)}$, $l^{\beta}_{\mu(t)}$), esta função devolve dois vetores ω^1 e ω^2.
- Em seguida atualizamos os *payoffs* sintéticos. Tal procedimento é realizado a cada instante $t \leq T_{\max}$.
- Com os vetores ω^1, ω^2, V_1 e V_2 atualizados obtemos cada elemento das matrizes M_1' e M_2'. Cabe ressaltar que esse procedimento é o mesmo utilizado para obtenção dos elementos das matrizes onde é definido o processo de escolha das estratégias.
- Os números obtidos em ω^1 e ω^2 são submetidos a um filtro dos limitantes l^{α} e l^{β}, obtendo-se assim dois outros vetores ω^1 e ω^2 ajustados por esse filtro.
- Com os vetores ω^1 e ω^2 criamos uma tabela com janelas de tempo, pois a simulação ocorre no tempo $T_{\max}$. Assim, teremos vários ω^1_{μ} e ω^2_{μ}.
- Dentre esse processo da simulação temos ainda uma busca do melhor do conjunto de vetores Ω^1_{μ} e Ω^2_{μ}, finalmente nesses vetores resultarão os elementos que formarão a matriz M_1' e M_2'.

4 Resultados

Os algoritmos foram executados sem o conhecimento das matrizes originais (M_1 e M_2) conhecendo-se somente os parâmetros $A(t)$ e $\mu(t)$.

A seguir apresentamos um gráfico comparativo da distância de Hamming entre M_1 e M_2 e as médias de M_1' e M_2', veja a figura 1 (esquerda). A eficiência do algoritmo é medida por baixos valores da distância de Hamming. Assim, de acordo com a figura 1 (esquerda) o algoritmo PLI0-1 tem melhor desempenho ao longo de α em relação ao AG. Vale ressaltar que para valores baixos de α a diferença de desempenho é pequena e a flutuação é alta. Analisando a figura (1) (esquerda) nota-se que os Erros Quadráticos Médios em relação a α apresentam-se com dois comportamentos diferentes. Primeiramente, o AG tem menores valores para α pequenos e ocorre uma inversão a partir de $\alpha \approx 1.5$ onde o PLI0-1 é definitivamente melhor.

DOI: 10.5540/03.2017.005.01.0418

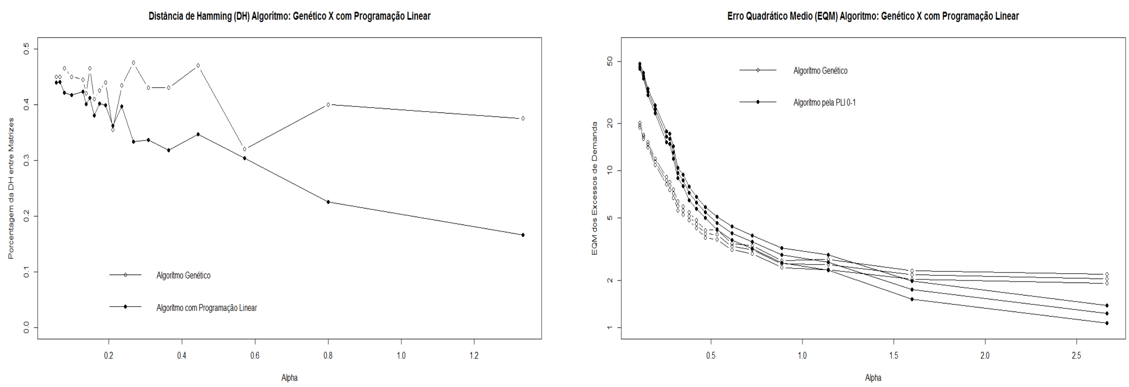

Figura 1: Gráfico da esquerda: os valores mínimos das distâncias de Hamming entre matrizes na otimização via AG (pontos abertos) e otimização via PLI0-1 (pontos fechados) em função de α. Gráfico da direita: representa o Erro Quadrático Médio com intervalo de confiança de 95% de confiança. Nota-se que o AG apresenta menores erros em relação ao algoritmo PLI0-1 para α pequeno e depois ocorre uma inversão para $\alpha > 1.5$. A linha com pontos abertos refere-se ao AG e com pontos fechados representam o PLI0-1.

Este resultado é intrigante pois sabemos da literatura [6,7] que, em relação a α, o jogo apresenta dois comportamentos distintos. Para valores pequenos de α o jogo está numa fase não ergódica e imprevisível ($H = 0$). Nessa fase, espera-se que ambos algoritmos sejam mais ineficientes. Na medida em que α cresce, passando por um valor crítico ($\alpha \approx 1$), inicia-se a fase ergódica e o jogo passa a ser mais preditível ($H > 0$). A região ergódica é aquela em que os algoritmos tem maiores chances de encontrar a solução ótima porque nessa região o comportamento coletivo não depende das condições iniciais, e o ótimo global fica melhor definido. Na região não ergódica tem muitos mínimos locais dificultando o trabalho das heurísticas.

Uma outra explicação é que para valores baixos de α existem muitas estratégias repetidas no jogo. Assim sendo, há muitas combinações diferentes do uso de estratégia que produzem o mesmo resultado. Este cenário equivale a uma situação em que existem soluções degeneradas, portanto, falta informação para que se possa levantar tal degenerescência e favorecer a engenharia reversa. A degenerescência vai diminuindo quando se aumenta α, o que permite encontrar distâncias de Hamming menores.

Não se sabe demonstrar se é possível chegar à solução ótima na fase ergódica. O que podemos fazer é propor heurísticas cada vez mais eficientes para encontrar soluções cada vez mais próximas do ótimo. Este é certamente um problema desafiador para matemáticos e cientistas da computação.

4.1 Uma aplicação da engenharia reversa no Jogo de Minoria

Na Análise de funções booleanas queremos conhecer a estrutura de uma função a partir de suas propriedades. Uma delas, a *influência* de uma das variáveis, mede a sensitividade da função respeito as mudanças de valor dessa variável. A *energia* da função é a soma das influências das variáveis. Essas propriedades são expressas em função dos coeficientes

DOI: 10.5540/03.2017.005.01.0418

da transformada de Fourier da função (veja [8]). Obtemos através desse estudo o apontamento dos dois jogadores mais influentes e esses jogadores, agora denominados *jogadores infiltrados*, participam de um novo jogo, junto com os demais. Trata-se de uma informação privilegiada similarmente ao que ocorre no mercado de ativos reais. O objetivo foi avaliar o seus desempenhos comparando-se os resultados obtidos.

Realizamos a engenharia reversa para obtermos as duas matrizes M_1' e M_2' conforme o algoritmos PLI0-1 e, na sequência, selecionamos em M_1' e M_2' as colunas correspondentes aos jogadores mais influentes. Agregamos essas duas colunas às matrizes M_1 e M_2 do jogo inicial, em seguida simulamos um jogo e constatamos que esses jogadores obtêm resultados de ganhos de energia, como vemos na figura 2.

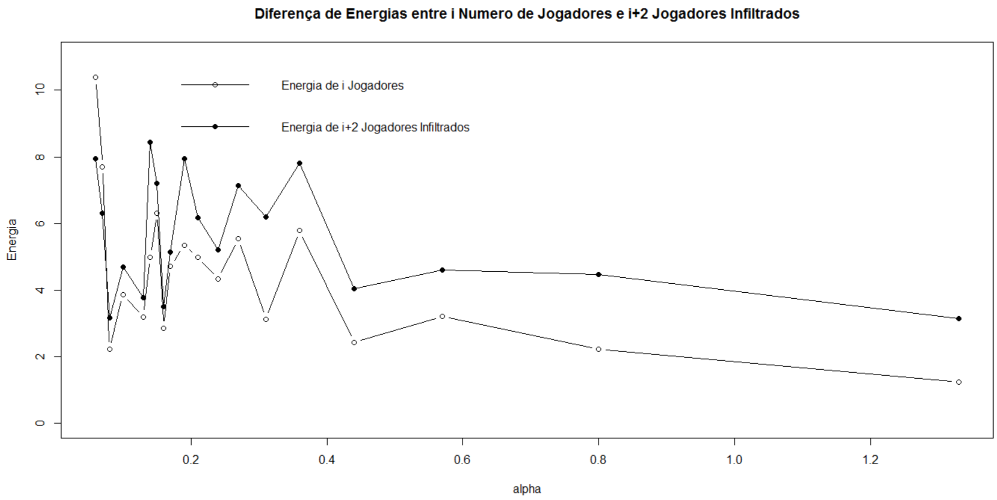

Figura 2: Energia com 2 jogadores infiltrados em função de α. O círculo aberto consiste no cálculo da energia (soma das influências) na simulação do jogo. A curva com pontos fechados representa a energia do jogo quando se introduz dois jogadores mais influentes que os demais.

5 Conclusões

Os resultados apontam o desempenho excelente do algoritmo baseado em Programação Linear Inteira Mista (PLI0-1) em contrapartida ao Algoritmo Genético (AG) considerando-se o menor erro possível no problema da engenharia reversa. Contudo, foi possível também simular o parâmetro A (excesso de demanda) cujos resultados se apresentam melhor em PLI0-1, na fase inicial, em detrimento do AG que supera em menor erro na fase final.

Ainda, foi possível estudar algumas propriedades interessantes das funções booleanas envolvidas no jogo, tais como a influência e a energia e, dessa forma, aplicar tais conceitos na simulação do Jogo de Minoria aferindo-se alguns comportamentos recorrentes dos jogadores. Essa foi a principal contribuição deste trabalho, acrescentamos novas e promissoras técnicas para o estudo da engenharia reversa, apresentado aqui na seção 3. Tais estudos deverão ser aprofundados em trabalho futuro.

DOI: 10.5540/03.2017.005.01.0418

A importância de se desenvolver algoritmos de engenharia reversa está na possibilidade de desvendar os mecanismos microscópicos subjacentes a dinâmicas de muitos agentes interagentes, ou seja, de sistemas complexos. Este conhecimento irá abrir uma nova área do conhecimento que é a engenharia de sistemas complexos.

Referências

[1] Damien Challet, Alessandro Chessa, Matteo Marsili, and Yi-Cheng Zhang. From minority games to real markets. *Quantitative Finance*, 1(1):168–176, 2001.

[2] Ying Ma, Guanyi Li, Yingsai Dong, and Zengchang Qin. Minority game data mining for stock market predictions. In *Agents and Data Mining Interaction*, pages 178–189. Springer, 2010.

[3] David Lamper, Sam D Howison, and Neil F Johnson. Predictability of large future changes in a competitive evolving population. *Physical Review Letters*, 88(1):017902, 2001.

[4] J. Wiesinger, D. Sornette, and J. Satinover. Reverse engineering financial markets with majority and minority games using genetic algorithms. *Computational Economics*, 41(4):475–492, 2012.

[5] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, SFCS '88, pages 68–80, Washington, DC, USA, 1988. IEEE Computer Society.

[6] Antonio Fernando Crepaldi, Fernando Fagundes Ferreira, and José de Souza Rodrigues. Minority game: an agent-based model applied to financial market. *Gestão & Produção*, 19(4):793–809, 2012.

[7] Damien Challet, Matteo Marsili, and Yi-Cheng Zhang. *Minority games: interacting agents in financial markets.* Oxford University Press, 2013.

[8] Ryan O'Donnell. *Analysis of boolean functions.* Cambridge University Press, 2014.

DOI: 10.5540/03.2017.005.01.0418