DOI: 10.22481/intermaths.v2i2.9819
Licença Creative Commons
Artigo | Article

INTERMATHS
ISSN 2675-8318
Vol. 2, N. 2, Jul - Dez 2021, p. 59 – 74

Experimentos numéricos sobre o Método de Yuan para problemas de Equilíbrio de Nash

Luís Felipe Bueno

UNIFESP, São José dos Campos-SP, Brasil

lfelipebueno@gmail.com

Amanda Gomes Vetorazzi

ITA/UNIFESP, São José dos Campos-SP, Brasil

agomesvetorazzi@gmail.com

Numerical experiments on Yuan's Method for Nash equilibrium problems

Abstract

In this paper numerical results are present on procedures to identify Nash Equilibrium. These experiments were systemized aiming to compare classical approaches with the algorithm propose by Y. Yuan in 2011. The method introduced by Yuan uses a Jacobi-type algorithm and a trust region globalization, developed specifically for Nash Equilibrium problems. In the original work, a theorical analysis is provide, but no considerations on the algorithm's practical performance are made. Therefore, this paper goal is to meet the need of a numerical implementation and to discuss the algorithm's numerical behave. In order to do that, the methods were tested in six different examples involving two-players dynamics. The examples differ from each other by the objective function of each player, concerning to vary the theoretical characteristics of them, which is the focus of this analysis. Finally, the effectiveness of both methods in finding a solution for Nash Equilibrium Problems is compared and important points, that influences the performance of Yuan's algorithm, are raised.

Keywords: Nash Equilibrium, Jacobi-type methods, Trust region, Numerical implementation.

MSC: 97N40; 97M40; 91A05.

Resumo

Neste artigo são apresentados resultados numéricos sobre procedimentos para identificação do Equilíbrio de Nash. Esses experimentos foram sistematizados com o objetivo de comparar algumas abordagens clássicas com o algoritmo proposto por Y. Yuan em 2011. O método intorduzido por Yuan conta com um algoritmo do tipo Jacobi e globalização com região de confiança desenvolvida especificamente para problemas de Equilíbrio de Nash. No trabalho original, uma análise teórica é disponibilizada, mas não são feitas considerações sobre o desempenho prático do algoritmo. Desta forma, o objetivo desse artigo é suprir a necessidade de implementação e discussão do comportamento numérico do algoritmo. Para isto, os métodos foram testados em seis exemplos numéricos diferentes envolvendo dinâmicas entre dois jogadores. Os exemplos se diferenciam entre si pela função objetivo de cada jogador, de forma a variar as características teóricas de cada problema, sendo este o foco de análise. Por fim, a eficácia dos métodos em encontrar uma solução para problemas de Equilíbrio de Nash foi comparada e apresentamos pontos importantes que influenciam o desempenho do algoritmo proposto por Yuan.

Palavras-chave: Equilíbrio de Nash, Métodos do tipo Jacobi, Região de confiança, Implementação numérica.

Submetido em: 01 de novembro de 2021 – **Aceito em**: 18 de dezembro de 2021

1 INTRODUÇÃO

Algoritmos para Equilíbrio de Nash têm despertado o interesse de pesquisadores em diversas áreas do conhecimento, sejam estas voltadas para engenharia, economia, pesquisa operacional, finanças e outras. Acredita-se que esse interesse esteja intimamente relacionado com a investigação de dinâmicas competitivas, sendo estas frequentes nas áreas mencionadas. Outro motivo que leva ao estudo desse tema são os interessantes problemas de otimização interligados entre si originados deste tipo de situação. Um problema de Equilíbrio de Nash padrão, do inglês *Nash Equilibrium Problem* (NEP), é definido como uma situação de conflito não cooperativa entre dois ou mais agentes tomadores de decisão, que possuem como propósito otimizar funções objetivo individuais, mas que são dependentes das escolhas dos demais participantes da dinâmica. São denominados jogos não cooperativos interações em que não há possibilidade de colaboração entre os jogadores, diferentemente de jogos cooperativos, nos quais é possível que os participantes negociem escolhas, assim como apresentado em [1]. Outra possibilidade para abordagem desse tema são os NEP generalizados, em inglês *Generalized Nash Equilibrium Problems* (GNEP), nos quais, além das funções objetivo, o conjunto de estratégias dos jogadores também é dependente da estratégia dos demais. Em [2] e [3] são apresentados apanhados extensos sobre diversos algoritmos e aplicações para essa classe de problemas.

Em um jogo, o conjunto de estratégias que caracteriza um Equilíbrio de Nash é o conjunto de soluções que otimiza o interesse de cada participante, quando as escolhas dos demais jogadores é fixada [4]. Para um NEP de 2-jogadores, como os que serão exibidos nesse trabalho, cada jogador $v \in \{1, 2\}$ busca minimizar uma função objetivo u_v controlando um conjunto de variáveis x_v pertencente ao conjunto de estratégias $\mathbb{R}^{n_v}$. Os casos em que $n_v > 1$ indicam que a estratégia x_v do jogador v contém mais de uma variável. Considera-se então que $n = \sum_v^2 n_v$ é o total de variáveis que integram a dinâmica. Para a função $u_v : \mathbb{R}^n \to \mathbb{R}$, define-se x como o vetor das decisões de ambos os jogadores em que

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^n.$$

Com a finalidade de evidenciar a informação sobre qual estratégia é controlada pelo jogador v, é conveniente que x seja representado pela participação $x = (x_v, \mathsf{x}^{-v})$. Essa decomposição indica por x^{-v} as variáveis não controladas pelo jogador v. Com isso, a iteração entre dois jogadores consiste em resolver dois problemas de otimização, em que um ponto x^* é dito um Equilíbrio de Nash quando a estratégia $(x^*)_v$ é solução para o problema de otimização

$$\min_{x_v \in \mathbb{R}^{n_v}} u_v(x_v, (\mathsf{x}^*)^{-v}). \tag{1}$$

Resolver (1) significa que, no instante em que um jogador, oponente de v, escolhe a estratégia $(\mathsf{x}^*)^{-v}$, o jogador v escolhe $(x^*)_v$ como a melhor resposta a esta escolha. Quando isso acontece para todo v, o ponto $x^* = ((x^*)_v, (\mathsf{x}^*)^{-v})$ é tido como um Equilíbrio de Nash, pois nenhum jogador tem incentivo para fazer uma escolha diferente a atual. Em muitos casos, um Equilíbrio de Nash é decorrente de um processo iterativo em que os jogadores, após uma sequência de escolhas simultâneas, estabilizam em um ponto estacionário. Desta forma, algoritmos com esta filosofia evocam um sentido especial em que o desenvolvimento numérico reflita um comportamento prático. Em [5], tem-se uma recente revisão sobre como algoritmos de melhor resposta vêm sendo explorados para uma abordagem iterativa de dinâmicas variadas. Sucintamente, a estratégia de melhor resposta é a decisão que otimiza a função objetivo do jogador para um conjunto fixo de escolhas dos demais jogadores. Desta forma, uma busca pela solução pode ser feita de forma iterativa onde, a cada instante, os jogadores tomam decisões individuais como a melhor resposta à decisão dos demais jogadores no instante anterior.

Um ponto ótimo deve satisfazer as condições necessárias de primeira ordem. Com isso, a resolução de um problema de Equilíbrio de Nash muitas vezes é realizada de forma a resolver o sistema resultante destas condições. Considerando uma dinâmica com dois jogadores $v \in \{1, 2\}$ em que a decisão do oponente é fixada, então a solução x_v deve ser tal que o gradiente da função u_v, em relação a variável de controle, seja nulo. Essa afirmação implica que ambos os jogadores escolhem x_v que resolva a equação

$$g_v(x) \equiv \frac{\partial u_v(x_v, \mathsf{x}^{-v})}{\partial x_v} = 0.$$

Como essa condição deve ser satisfeita para os dois jogadores, obtemos um sistema que é denotado como se segue

$$F(x) = \begin{pmatrix} g_1(x_v, \mathsf{x}^{-v}) \\ g_2(x_v, \mathsf{x}^{-v}) \end{pmatrix} = 0. \tag{2}$$

Diversos métodos podem ser utilizados para a resolução do Sistema (2). No caso em que o Sistema (2) contenha equações não lineares, certamente o método mais consagrado é o método de Newton, que resolve sistemas não lineares através de uma sequência de aproximações lineares. Ou seja, dada uma estimativa de solução x^k, uma nova estimativa é obtida de forma que

$$F(x^k) + J(x^k)(x^{k+1} - x^k) = 0, \tag{3}$$

onde $J(x)$ é o jacobiano da função $F(x)$.

Como F é formado pelos gradientes das funções de cada jogador, essas aproximações lineares podem ser obtidas encontrando pontos estacionários de aproximações quadráticas de u_v. Este procedimento é bastante explorado na resolução de problemas de otimização [7] e apresenta fortes relações com o Algoritmo de Yuan, apresentado na próxima seção. A interpretação proveniente de modelos quadráticos de funções objetivo ajuda a agregar elementos de minimização no processo, evitando vários pontos estacionários que não são soluções do sistema. Até onde sabemos, o algoritmo de Yuan é o único que explora estes aspectos na resolução de NEPs. De modo geral, é muito comum nos trabalhos de NEPs que sejam usadas hipóteses de convexidade para eliminar as dificuldades acerca de pontos estacionários que não são minimizadores. Ao evitar este tipo de hipótese, o método de Yuan pode ser aplicado em uma gama muito maior de problemas.

Outro fato importante sobre o método de Newton é que, sob certas condições, o método possui convergência local quadrática. Isso significa que ele é bem eficiente perto de uma solução. Entretanto, para garantir convergência a partir de um ponto arbitrário, são necessárias técnicas de globalização que podem diminuir o tamanho do passo de modo a exigir que $\|F(x)\|$ seja reduzida a cada iteração, conforme descrito em [7]. Especificamente para métodos de otimização, existem técnicas de globalização que medem o progresso obtido não pelo valor do resíduo das condições de primeira ordem, mas sim pela função objetivo do problema. Novamente, esta é uma estratégia que beneficia a obtenção de minimizadores, em vez de meros pontos estacionários. Como em um NEP existem mais de uma função objetivo, não é imediato encontrar uma maneira de medir se está havendo progresso em uma iteração. O algoritmo de Yuan usa uma estratégia que combina tanto reduções de u_v quanto de F, aproveitando elementos da estrutura de minimização.

A resolução do sistema (2) também pode ser feita por meio de um processo associado à dinâmica iterativa de melhor resposta. Fixada a decisão do oponente em um dado instante, que denotaremos por $(\mathrm{x}^k)^{-v}$, para obter a solução ótima do jogador v, devemos obter x_v^{k+1} que anule $g_v(x_v^{k+1}, (\mathrm{x}^k)^{-v})$. Sob hipóteses de convexidade, esta estratégia seria equivalente à de melhor resposta, mas não necessariamente no caso geral. Se $F(x) = 0$ é um sistema linear, escrito na forma $Ax-b$, esta estratégia coincide com o tradicional método de Jacobi por blocos. Este é um dos algoritmos mais clássicos para resolução de sistemas lineares e usa parte das equações para encontrar uma estimativa de parte das variáveis, fixando os valores das demais. Analiticamente, para blocos de tamanho um, é considerada a decomposição da matriz dos coeficientes em $A = L+D+R$, onde L é a parte triangular estritamente inferior de A, D é formada pela diagonal de A e U por sua parte triangular estritamente superior. Da partição de A, é possível obter $M = -D^{-1}(L + R)$, chamada de matriz iterativa do método, de modo que

$$x^{k+1} = Mx^k + D^{-1}b.$$

Um resultado clássico, veja por exemplo [6], diz que é possível garantir a convergência para a solução de $F(x) = 0$, independente de x^0 sempre que $\|M\| < 1$. Uma condição suficiente para isto é que A seja diagonalmente dominante, ou seja, o módulo de cada elemento da diagonal é maior do que a soma dos módulos dos outros elementos que estão em sua linha. Outra condição suficiente, para o caso 2×2, é que a matriz A seja simétrica e definida positiva, ou seja $A = A^T$ e $x^T Ax > 0$ para todo x não nulo.

Ainda do caso em que F é um sistema não linear, nos referimos a métodos do tipo Jacobi como sendo aqueles em que as escolhas para nova estimativa de solução de (2) no instante $k+1$ são feitas em paralelo, ou seja, simultaneamente para todo jogador, fixando as estratégias do instante k. A solução de cada bloco do sistema, fixada parte das variáveis, não é imediata. Uma possibilidade neste sentido é considerar métodos de Newton-iterativo, como os apresentados por [8]. Dentre esses métodos, é possível que a configuração de parâmetros proporcione um método do tipo Jacobi, formulado com base em iterações de Richardson-D'Jakonov. Esta opção nada mais é do que fazer uma iteração do método de Jacobi para resolver o sistema do método de Newton e já trocar a estimativa de solução.

Por fim, o algoritmo de Yuan que será introduzido a seguir, combina características do tipo Jacobi, com modelos quadráticos e uma estratégia de globalização que leva em consideração tanto o decréscimo nas funções objetivo de cada jogador quanto do resíduo do sistema das condições de primeira ordem como um todo. Embora uma análise teórica cuidadosa do algoritmo seja apresentada no artigo original de Yuan [9], não são reportados resultados numéricos do algoritmo. Nossa contribuição aqui é apresentar uma implementação do algoritmo e discutir alguns aspectos de seu comportamento prático. Pelas características apresentadas na construção do algoritmo, vamos compara-lo com métodos do tipo Jacobi para resolução de sistemas e, para um exemplo onde F é não linear, com uma filosofia do tipo Newton.

2 PROCEDIMENTOS PARA RESOLUÇÃO PELO ALGORITMO DE YUAN

Uma possibilidade de algoritmo específico para NEP é proposta por Y. Yuan em [9]. O algoritmo de Yuan utiliza métodos do tipo Jacobi e regiões de confiança para identificação de estratégias que constituam um Equilíbrio de Nash. Nesse algoritmo são elaborados problemas de otimização auxiliares, modelados a partir de aproximações quadráticas, para a análise de cada participante. A cada problema auxiliar é atribuída uma restrição, denominada região de confiança, que tem o intuito de garantir que os resultados propostos pelos modelos simplificados sejam semelhantes aos resultados da função original. Em toda iteração a região de confiança é recalculada individualmente, dependendo da qualidade dos pontos de iteração atualizados, para os quais são estabelecidos novos problemas auxiliares. Por fim, a condição de parada do algoritmo está associada a condição de um ponto ser estacionário. Este ponto é declarado como sendo uma estimativa de Equilíbrio de Nash para o problema.

No que se segue, é apresentado uma simplificação do algoritmo proposto por Yuan, para o caso de problemas sem restrições e considerando a opção particular de escalonamento feito pela matriz identidade. Esse algoritmo considera sendo $u_v(x)$ a função objetivo do jogador v, $g_{v,k} = \frac{\partial u_v(x)}{\partial x_v}|_{x=x^k}$ o gradiente de u_v em relação a x_v avaliado em x^k, e $B_{v,k} = \frac{\partial^2 u_v(x)}{\partial x_v^2}|_{x=x^k}$ uma aproximação da Hessiana. Em posse destas informações, define-se o problema auxiliar para cada jogador

$$\begin{aligned} \text{min } & \phi_{v,k}(d_v) = u_v(x^k) + d_v^T g_{v,k} + \frac{1}{2} d_v^T B_{v,k} d_v \\ \text{s. a } & ||d_v||_2 \leq \Delta_{v,k} \\ & (x^k)_v + d_v \in \mathbb{R}^{n_v}, \end{aligned} \tag{4}$$

que representa uma aproximação quadrática do problema original. A solução de (4) é denotada pela variável $d_{v,k}$. Essa variável indica uma expectativa de mudança em $(x^k)_v$ para melhorar a função objetivo do jogador v na k-ésima iteração. Esses problemas auxiliares possuem como restrição uma região de confiança

$$\Delta_{v,k} = \frac{1}{\tau_v + t_{v,k}} ||g_{v,k}||_2, \tag{5}$$

que busca garantir que a representação do modelo simplificado $\phi_{v,k}(d_v)$ seja localmente semelhante a função original $u_v(x)$. O componente $\Delta_{v,k}$ é formulado por uma parcela do gradiente da função u_v, e limita as mudanças em x_v a uma região de confiança. O parâmetro positivo τ_v é mantido constante em todas as iterações, enquanto $t_{v,k} \geq 0$ é ajustado a depender da eficiência da iteração. Sendo assim, em toda iteração essa região é atualizada individualmente, estabelecendo novos problemas auxiliares.

A aceitação ou rejeição da mudança $d_{v,k}$, sugerida pelo subproblema, é definida pela razão entre a redução real, $Ared$, e a redução prevista pelo modelo aproximado, $Pred$. Isto é feito definindo

$$r_{v,k} = \frac{Ared_{v,k}}{Pred_{v,k}} = \frac{u_v((x^k)_v, (x^k)^{-v}) - u_v((x^k)_v + d_{v,k}, (x^k)^{-v})}{\phi_{v,k}(0) - \phi_{v,k}(d_{v,k})},$$

e o ponto da iteração seguinte como

$$(x^{(k+1)})_v = \begin{cases} (x^k)_v + d_{v,k}, \text{ se } r_{v,k} > 0, \\ (x^k)_v, \text{ caso contrário.} \end{cases} \quad (6)$$

Observe que $r_{v,k}$ não está bem definido se $\phi_{v,k}(0) = \phi_{v,k}(d_{v,k})$. Nestes casos prosseguimos o algoritmo considerando que $r_{v,k} > 0$, pois acreditamos que a solução do modelo seja de boa qualidade, pelos motivos que discutidos a seguir. Note que, da maneira como $\Delta_{v,k}$ é calculado em (5), tem-se que $\Delta_{v,k} = 0$ se e somente se $g_{v,k} = 0$. Nestes casos já estamos em ponto estacionário de $u_v(x_v, (x^k)^{-v})$ e portanto é natural declararmos que a solução nula seja uma boa atualização.

Consideremos agora que $\Delta_{v,k} > 0$. Como $d_{v,k}$ é ponto de mínimo da função $\phi_{v,k}$, e $d_{v,k} = 0$ é um ponto viável para (4), então a diferença $\phi_{v,k}(0) - \phi_{v,k}(d_{v,k})$ é sempre não negativa. Logo, se $d_{v,k} = 0$ é uma solução ótima para (4), vale a igualdade

$$\nabla \phi_{v,k}(0) = 0.$$

Note que $\nabla \phi_{v,k}(d_v) = g_{v,k} + Bd_v$. Portanto, $\nabla \phi_{v,k}(0) = 0$ implica que $g_{v,k} = 0$. Sendo assim, $(x^k)_v$ é um ponto estacionário de $u_v((x^k)_v, (x^k)^{-v})$ e novamente temos que é razoável declarar que tenhamos obtido uma boa atualização. O fato de $Pred_{v,k} \geq 0$ implica que, a não ser no caso de $d_{v,k} = 0$, o passo é aceito sempre que $u_v((x^k)_v, (x^k)^{-v})$ reduzir.

Tendo estipulado o próximo ponto de iteração, o parâmetro $t_{v,k}$ do algoritmo é atualizado a partir da função de mérito $\psi(x)$, que combina a medida de estacionaridade dos dois problemas, calculada como

$$\psi(x) = ||F(x)||^2 = \sum_{v=1}^{2} ||g_{v,k}||^2,$$

e portanto, a redução desta sugere sucesso na busca pela solução.

Considerando $\eta_k = \min_{1 \leq i \leq k} \psi(x_i)$, define-se o parâmetro

$$\rho_k = \frac{\eta_k - \psi(x_{k+1})}{Pred_k},$$

em que $Pred_k = \sum_{v=1}^{2} Pred_{v,k}$. Note que $Pred_k = 0$ tem como consequência x^k ser um ponto estacionário do PEN, para o qual não se espera uma nova iteração. O valor de ρ_k é determinado pelo progresso obtido na redução de ψ em relação ao melhor ponto já obtido, proporcionalmente ao valor de $Pred_k$.

O critério que determina se a composição de modelos simplificados é uma boa aproximação para representar a dinâmica original consiste em verificar se ρ_k é suficientemente grande. Caso $\rho_k \geq \beta_1$ então os modelos aproximados reduzem satisfatoriamente as condições de primeira ordem do problema original. Sendo assim, os parâmetros $t_{v,k}$ são atualizados individualmente de acordo com o valor $r_{v,k}$ associado a cada jogador, nivelado por β_2, tal que

$$t_{v,k+1} = \begin{cases} \max[t_{v,k} - \delta_v, 0], & \text{se } r_{v,k} \geq \beta_2; \\ t_{v,k}, & \text{se } r_{v,k} \in (0, \beta_2]; \\ t_{v,k} + \delta_v, & \text{se } r_{v,k} \leq 0. \end{cases} \tag{7}$$

Caso $\rho_k < \beta_1$, então, conclui-se que os modelos simplificados não representam bem a dinâmica original como um todo, requisitando que todas as regiões de confiança diminuam na próxima iteração, de modo que

$$t_{v,k+1} = t_{v,k} + \delta_v, \qquad \forall v \in \{1,2\}. \tag{8}$$

Por fim, havendo atualizado os componentes, $t_{v,k+1}$ por (7) ou (8), e $\Delta_{v,k+1}$ por (5), a condição de parada do algoritmo está associada a um ponto em que $\psi(x) = 0$. Na prática o processo iterativo do algoritmo é interrompido se ψ atinge um valor menor do que uma tolerância dada.

O Algoritmo

(P0): Defina $k := 1$ e um ponto inicial factível $x^{(1)} \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$.
Escolha constantes positivas τ_v, δ_v, $t_{v,1}$ $\forall v \in \{1,2\}$.
Escolha $\beta_1 \in (0,1), \beta_2 \in (0,1)$.
Calcule $\Delta_{v,1}$ $\forall v \in \{1,2\}$.

(P1): Se $\sum_{v=1}^{2} ||g_{v,k}||_2 = 0$ então pare.
Senão resolva (4) para obter $d_{v,k}$ para todo v.

(P2): Compute $r_{v,k}$.
Defina o próximo ponto da iteração x^{k+1}.

(P3): Atualize $t_{v,k}$ e $\Delta_{v,k}$

(P4): Defina $k := k+1$ e vá para o Passo 1.

A maneira como Δ é atualizado é possivelmente a contribuição mais significante do algoritmo analisado. A princípio existem diversas maneiras para que o valor dessa região de confiança seja determinado, e podem ser conferidas em [10] e [11]. Isso posto, o cálculo de Δ no algoritmo de Yuan é interessante para um PEN por dois motivos. O primeiro motivo envolve a avaliação individual de cada jogador, através da atualização de $t_{v,k}$. Isso inclui considerar informações exclusivas do participante na dinâmica, de modo que o jogador tenha o problema auxiliar elaborado de forma personalizada. O segundo motivo é que essas personalizações estão relacionadas à função objetivo do jogador, mais especificamente ao gradiente desta. Essa abordagem proporciona um cálculo da região de confiança com significado tanto para cada jogador, quanto para a dinâmica como um todo. Ademais, essa forma de atualização origina o Teorema (2.1), enunciado de maneira adaptada para o caso considerado neste artigo.

Teorema 2.1 *Considere que $u_v(x)$, $v \in \{1,2\}$, sejam duas vezes continuamente diferenciável e denote por $J(x)$ a matriz Jacobiana de $F(x)$. Se $J(x)$ é uniformemente definida positiva e $B_{v,k}$ uniformemente limitada para todo v, k, e $x \in \mathbb{R}^n$, então os pontos de iteração $\{x_k\}$ gerados pelo algoritmo são tais que*

$$\lim_{k\to\infty} \eta_k = 0. \tag{9}$$

3 EXPERIMENTOS NUMÉRICOS

A seguir, são apresentados seis experimentos numéricos, em dinâmicas com dois jogadores, nas quais cada jogador faz apenas uma escolha no conjunto dos reais. Os três primeiros exemplos são compostos por funções polinomiais quadráticas semelhantes entre si, diferenciadas por variações no coeficiente que acompanha o termo ao quadrado. O quarto exemplo é elaborado com funções objetivo cúbicas, que permite estudar o caso que a matriz Hessiana não é constante. Para os dois últimos experimentos, buscou-se exemplos contextualizados de forma a motivar a compreensão da teoria em situações cotidianas.

Exemplo 1 *Nesse primeiro exemplo considera-se a dinâmica*

$$\textit{Jogador 1: } \min_{x_1\in\mathbb{R}} u_1(x_1,x_2) = x_1^2 + x_1x_2 - 5x_1,$$
$$\textit{Jogador 2: } \min_{x_2\in\mathbb{R}} u_2(x_1,x_2) = \frac{3x_2^2}{2} - x_1x_2 - x_2.$$

Neste caso ambas as funções objetivo possuem apenas um ponto de mínimo para cada estratégia do oponente, o que significa uma única melhor resposta associada a escolha do adversário. Além disso, ao avaliar o sistema de equações de melhor resposta é possível perceber que o tamanho da matriz iterativa $||M||$ é menor do que um, garantindo a convergência do método de Jacobi aplicado ao sistema linear com equações de melhor resposta.

Exemplo 2 *Nesse segundo exemplo, consideramos*

$$\textit{Jogador 1: } \min_{x_1\in\mathbb{R}} u_1(x_1,x_2) = \frac{x_1^2}{4} + x_1x_2 - 5x_1,$$
$$\textit{Jogador 2: } \min_{x_2\in\mathbb{R}} u_2(x_1,x_2) = \frac{x_2^2}{6} - x_1x_2 - x_2.$$

Novamente, ambas as funções objetivo também possuem apenas um ponto de mínimo para cada estratégia adversária, porém ao avaliar o sistema de equações de melhor resposta, tem-se que $||M|| \geq 1$. Com essas informações, nada se pode concluir sobre a convergência do método de Jacobi para o sistema linear de melhor resposta, embora o Equilíbrio de Nash exista para essa dinâmica.

Exemplo 3 *Nesse terceiro exemplo, temos o jogo*

$$\begin{aligned} &\textit{Jogador 1: } \min_{x_1 \in \mathbb{R}} \; u_1(x_1, x_2) = x_1^2 + x_1 x_2 - 5x_1, \\ &\textit{Jogador 2: } \min_{x_2 \in \mathbb{R}} \; u_2(x_1, x_2) = -\frac{3x_2^2}{2} - x_1 x_2 - x_2. \end{aligned}$$

Note que o Jogador 2 tem como objetivo minimizar uma função sem ponto de mínimo, independentemente da decisão do Jogador 1. Isso significa que para toda estratégia do Jogador 1, o Jogador 2 tem melhor resposta indefinida. Ainda assim, é possível obter solução via Jacobi das condições de 1ª ordem, pois $||M|| < 1$, entretanto, o ponto estacionário encontrado não é solução para dinâmica.

Exemplo 4 *Nesse quarto exemplo, consideramos a dinâmica com funções objetivo não quadráticas dada por*

$$\begin{aligned} &\textit{Jogador 1: } \min_{x_1 \in \mathbb{R}} \; u_1(x_1, x_2) = \frac{x_1^3 x_2^2}{3} + \frac{x_1^2}{2}, \\ &\textit{Jogador 2: } \min_{x_2 \in \mathbb{R}} \; u_2(x_1, x_2) = \frac{x_1^2 x_2^3}{3} + \frac{x_2^2}{2}. \end{aligned}$$

Neste caso a função objetivo dos jogadores possui pontos de mínimos locais e máximos locais, em que o sistema de melhor reposta é composto por equações não lineares. Por este motivo, para resolver $F(x) = 0$ são necessárias técnicas para resolução de sistemas não lineares.

Os exemplos contextualizados que seguem são dinâmicas de soma zero, em que o objetivo de um jogador é minimizar uma função $U(x)$, enquanto o objetivo do oponente é maximizar essa mesma função.

Exemplo 5 *Esse exemplo se passa no contexto da cobrança de pênaltis, uma etapa decisiva em jogos de futebol. Essa situação é composta por dois jogadores em conflito; o Goleiro e o Atacante. Para estes jogadores, devido a agilidade em que se deve tomar a decisão, é sugerido a cada jogador que a escolha entre os lados do gol para o qual irão se direcionar, seja feita antes da cobrança. Se ambos os jogadores se encaminharem para sentidos opostos, as chances do gol acontecer é maior, mesmo* *que não certa. Por outro lado, caso os jogadores se dirijam para um mesmo sentido as chances de gol diminuem. Com isso, a Tabela (1) representa as chances de gol quando combinadas as escolhas de estratégias dos jogadores. Sabe-se ainda que o goleiro pula para Esquerda com probabilidade q, ou para o Direita com probabilidade $1-q$. Semelhantemente, o atacante chuta para o Esquerda com probabilidade p, ou chuta para Direita com probabilidade $1-p$. Essa tabela é uma adaptação para o jogo de futebol, do estudo de caso apresentado em [12, Cap. 8, p. 264].*

Tabela 1: Chances de gol

Atacante/ Goleiro	Esquerda p	Direita $(1-p)$
Esquerda q	50%	80%
Direita $(1-q)$	90%	20%

Fonte: Elaborada pelos autores.

Com essas informações, o Exemplo 5 é elaborado de modo que

$$\begin{array}{ll} \text{Atacante:} & \min\limits_{p\in\mathbb{R}} u_1(p,q) = -p(0,6-q), \\ \text{Goleiro:} & \min\limits_{q\in\mathbb{R}} u_2(p,q) = q(0,7-p). \end{array}$$

Exemplo 6 *Esse exemplo é inspirado em um exercício do livro [13] e compreende a interação entre o Governo, que deseja escolher uma porcentagem de distribuição de dois tipos de vacinas para uma certa doença; e um Vírus, que pode ocorrer sob a forma de duas variantes. Nesse jogo, assume-se o pressuposto de que toda população será vacinada por uma das vacinas e contrairá uma variação do vírus certamente. Essas suposições delimitam a avaliação da dinâmica, em considerar como fator principal, a eficácia de cada tipo de vacina contra cada sorotipo do vírus. Além disso, o exemplo ilustra uma dinâmica estritamente não cooperativa, na qual a possibilidade de negociação entre os participantes é impossível. Como informação complementar, é mencionado que a proporção em que dois sorotipos do vírus ocorrem na população é desconhecida. Por isso, foram desenvolvidas duas vacinas que possuem efeitos diferentes para cada tipo de vírus. Sabe-se que a eficácia da vacina 1 contra o vírus de sorotipo 1 é de 85%, enquanto contra o vírus de sorotipo 2 é de 70%. Semelhantemente, a eficácia da vacina 2 contra o vírus sorotipo 1 é de 60%, e contra o sorotipo 2 de 90%. O estudo para este exemplo tenta auxiliar a decisão do governo sobre qual deve ser política de vacinação, objetivando maximizar a eficácia da vacina. A Tabela (2) organiza as informações descritas.*

Tabela 2: Eficácia da vacina

Vírus/ Governo	Sorotipo 1 p	Sorotipo 2 $(1-p)$
Vacina 1 q	85%	70%
Vacina 2 $(1-q)$	60%	90%

Fonte: [13, Cap. 4, p. 97].

Seja p a variável que indica a probabilidade de ocorrência do vírus sorotipo 1, $(1-p)$ a probabilidade de ocorrência do vírus sorotipo 2, q a probabilidade de ocorrência da vacina 1 e $(1-q)$ a probabilidade de ocorrência da vacina 2. É possível estabelecer uma função que calcula as chances de um indivíduo se manter saudável dado ter contraído um sorotipo de vírus e ter tomado uma determinada vacina. Essa função é modelada a partir da esperança de eficácia da vacina como

$$U(p,q) = 0,85pq + 0,6p(1-q) + 0,7(1-p)q + 0,9(1-p)(1-q), \tag{10}$$

em que o primeiro termo se refere a tomar a vacina 1 e contrair o vírus sorotipo 1, o segundo termo a tomar a vacina 2 e contrair o vírus sorotipo 1 e assim sucessivamente. Apesar de p e q serem valores associados ao conceito de probabilidade, a função $U(p,q)$ está definida para todo conjuntos dos reais, e não necessariamente restrita ao intervalo $[0,1]$. Além disso, as funções objetivo da dinâmica podem ser representadas apenas pelos termos que acompanham as coordenadas controladas pelo jogador, já que os termos constantes, ou que dependam exclusivamente da decisão do oponente, não podem ser alterados pela escolha do jogador. Dessa forma, tem-se que o Vírus deseja minimizar (10), enquanto o Governo deseja maximizar essa mesma função e a interação entre esses agentes é representada por

$$\begin{aligned} &\text{Vírus:} && \min_{p\in\mathbb{R}} u_1(p,q) = p(0,45q-0,3),\\ &\text{Governo:} && \min_{q\in\mathbb{R}} u_2(p,q) = -\,q(0,45p-0,2). \end{aligned}$$

3.1 Resultados numéricos

A implementação numérica de todos os seis exemplos foi realizada utilizando as duas alternativas de procedimentos apresentadas, sendo a primeira através da resolução de sistemas que localiza pontos estacionários, podendo representar equações de melhor resposta. A segunda implementação utiliza o algoritmo. Em todos os casos, foi atribuído um ponto inicial $x^1 = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$. Para o método com resolução de sistemas, esta informação é suficiente para iniciar resolução. No experimento com o algoritmo de Yuan, considerou-se o conjunto de parâmetros iniciais

$$C_1 = \left\{ \delta_v = 0,01; \tau_v = 1; t_{v,1} = 1; \beta_1 = \beta_2 = \frac{1}{2} \right\}$$

para todo $v \in \{1, 2\}$. Também foi estabelecido como critério de parada um limite de 50 iterações, ou até que a soma em módulo dos gradientes assumisse um valor de tolerância próximo de zero, $tol = 10^{-5}$, sinalizando tendência a um ponto estacionário.

Os resultados do primeiro método de resolução são apresentados na Tabela (3) em que a eficácia é comparada com o segundo método de resolução que tem os resultados dispostos na Tabela (4). Considera-se que o método é eficaz quando este retorna ao menos uma tendencia ao Equilíbrio de Nash, dentro do limite de tolerância quando a existência deste é garantida. Caso o método retorne um ponto estacionário que não tenha característica de um Equilíbrio de Nash, então o procedimento é considerado ineficaz para resolução do exemplo.

Tabela 3: Experimento por resolução de sistemas via métodos do tipo Jacobi.

	Condição de Parada	Ponto Final	Equilíbrio de Nash	Eficácia
Exemplo 1	tol em 13 ite.	(2 ; 1)	(2 ; 1)	sim
Exemplo 2	nº max ite.	(3,51e+19 ; 6,29e+19)	(0,5714 ; 4,7142)	não
Exemplo 3	tol em 14 ite.	(3,2 ; -1,4)	não tem	não
Exemplo 4	tol em 7 ite.	(3,2760e-08 ; 3,2760e-08)	(0 ; 0)	sim
Exemplo 5	-	não definido	(0,6 ; 0,7)	sim
Exemplo 6	-	não definido	(0,6666 ; 0,4444)	sim

Fonte: Elaborada pelos autores.

Tabela 4: Experimento com Algoritmo de Yuan.

	Condição de Parada	Ponto Final	Equilíbrio de Nash	Eficácia
Exemplo 1	tol em 17 ite.	(2 ; 1)	(2 ; 1)	sim
Exemplo 2	nº max ite.	(0,5919 ; 4,5551)	(0,5714 ; 4,7142)	sim
Exemplo 3	nº max ite.	(-4284,7; 8821,1)	não tem	sim
Exemplo 4	tol em 18 ite.	(1,84e-05 ; 6,01e-06)	(0 ; 0)	sim
Exemplo 5	nº max ite.	(27,93 ; 358,74)	(0,6 ; 0,7)	não
Exemplo 6	nº max ite.	(-11,24 ; -1,94)	(0,6666 ; 0,4444)	não

Fonte: Elaborada pelos autores.

Começando a comparação pelo Exemplo 1, tem-se que ambos dos métodos foram eficazes em identificar o Equilíbrio de Nash da dinâmica. É provável que essa eficácia esteja relacionada à composição entre as funções objetivo, pois ambos os jogadores buscam minimizar funções com pontos de mínimos globais estritos.

Para o Exemplo 2 foi observada eficácia pela resolução através do algoritmo de Yuan, mas o mesmo não aconteceu para o experimento resolvendo o sistema de melhor resposta. A impossibilidade de encontrar uma solução por este método é devida ao fato de que o tamanho da matriz iterativa no método de Jacobi para sistemas lineares $||M||$ é menor do que um, não sendo possível estabelecer um processo iterativo convergente. Por outro lado, o algoritmo de Yuan demonstrou resolver dinâmicas com essa característica sem dificuldades.

No Exemplo 3, o resultado obtido pela resolução sistema não é um ponto com características de Equilíbrio de Nash, e por isso essa alternativa foi considera ineficaz para este problema. Repare que, supondo que a escolha de um dos jogadores pertença ao ponto final alcançado, tem-se que o outro jogador iria preferir uma escolha diferente para reduzir a função objetivo, e esse processo seria repetido indefinidamente para o caso sem restrições. Na realidade, isso é o que acontece com o algoritmo. Como a dinâmica não possui um Equilíbrio de Nash, o algoritmo realiza iterações indefinidamente. Esse resultado favorece as conclusões sobre o algoritmo, já que na ausência de uma solução, este não retorna um ponto estacionário qualquer, e por isso é considerado eficaz.

O Exemplo 4 se diferencia dos exemplos anteriores por possuir pontos de mínimos locais e pontos de máximos locais. Como o interesse na dinâmica é obter pontos de mínimos, espera-se que os métodos testados sejam capazes de perceber essa diferença. No caso em que a resolução consiste em resolver um sistema não linear os resultados, através do método de Newton-Jacobi para sistemas não lineares, são os dispostos na Tabela (3). A alternativa em resolver o sistema $F(x)$ pelo método de Newton-Jacobi, consiste em considerar o seguinte processo iterativo de Newton

$$J(x^k)x^{k+1} = F(x^k) - J(x^k)x^k. \tag{11}$$

mas utiliza apenas a primeira iteração da resolução do sistema (3), pelo método de Jacobi, como ponto de iteração seguinte. Para este caso o algoritmo de Newton-Jacobi obteve o ponto de Equilíbrio de Nash.

Para os demais exemplos, onde o sistema (2) é linear, o método de Newton teria encontrado o ponto estacionário em uma única iteração sendo ele um Equilíbrio de Nash ou não. Já neste exemplo, é interessante também verificar o comportamento do Método de Newton para o sistema (2). Aqui, embora resolver o sistema tenha sido considerado um método eficaz, é importante comentar que para pontos iniciais quaisquer é possível que o método de Newton não solucione a dinâmica adequadamente. Um exemplo disso é caso em que o método é iniciado por um ponto próximo ao ponto de máximo local, que então é retornado como solução para o sistema. Pelo algoritmo de

Yuan isso não acontece, pois a estrutura do algoritmo é capaz de diferenciar os pontos de interesse para resolução da dinâmica.

Quanto aos Exemplos 5 e 6, especulou-se sobre a possibilidade do algoritmo providenciar soluções para casos que não atendem as hipóteses estipuladas pelo Teorema (2.1). Isso acontece porque a matriz $J(x)$ é constante e da forma

$$\begin{pmatrix} 0 & a \\ -a & 0 \end{pmatrix}, \tag{12}$$

a qual não é positiva definida. Para estes exemplos o método de Jacobi para sistemas lineares nem estaria bem definido, pois a matriz diagonal é nula. Entretanto este tipo de problema aparece em várias aplicações básicas e portanto são relevantes. A implementação do algoritmo de Yuan, embora bem definida, também não conseguiu encontrar os pontos de equilíbrio. Notamos nestes casos uma situação interessante que a escolha de todo jogador depende integralmente da estratégia do adversário. Com isso, supondo que o Jogador 1 atinja o ponto de equilíbrio, seria necessário que a região de confiança do Jogador 2 fosse nula, para que ambos atingissem o equilíbrio. Esta observação nos leva a levantar possibilidades de alterações no algoritmo que envolva os dois problemas e suas condições de otimalidade na construção da região de confiança.

4 CONCLUSÃO

Apesar do artigo ser composto por exemplos específicos, para os quais não é possível fazer afirmações generalizadas, buscou-se através destes desenvolver intuição sobre a variedade de dinâmicas e como estas se comportam em diferentes métodos. Pode-se perceber que o método de Newton, que inclui a resolução de sistemas, encontrou pontos estacionários que não satisfazem a característica de um Equilíbrio de Nash, enquanto o método de Yuan evitou esses pontos. Em problemas onde não existe Equilíbrio de Nash, o método de Yuan não declarou convergência, o que se considera uma vantagem em relação a métodos que retornam soluções estacionárias, mas que não são Equilíbrio de Nash. Já em problemas em que existe Equilíbrio de Nash, mas não é possível identifica-lo por métodos do tipo Jacobi, em algumas vezes o algoritmo de Yuan obteve sucesso. Entretanto, a teoria do método de Yuan pode ser bastante restritiva, não contemplando o caso básico que aparece em diversas situações de problemas quadráticos de soma-zero, o que resultou na não convergência em problemas desse tipo. Como perspectivas futuras, estuda-se a possibilidade de ajustar o algoritmo analisado de modo a obter um resultado satisfatório em todos os casos. Outra possibilidade seria o estudo mais aprofundado sobre as condições de convergência dos métodos de resolução apresentados.

REFERÊNCIAS

[1] J. Nash, "Two-Person Cooperative Games", *Econometrica*, vol. 21, no. 1, pp. 128–140, 1953. https://doi.org/10.2307/1906951

[2] F. Facchinei e C. Kanzow,"Generalized Nash equilibrium problems", *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, 2010. https://doi.org/10.1007/s10479-009-0653-x

[3] A. Fischer, M. Herrich e K. Schönefeld, "Generalized Nash Equilibrium Problems - Recent Advances and Challenges", *Pesquisa Operacional*, vol. 34, no. 3, pp. 521–558, 2014. https://doi.org/10.1590/0101-7438.2014.034.03.0521

[4] J. Nash, "Non-cooperative games", *Annals of mathematics*, vol. 54, no. 2, pp. 286–295, 1951. https://doi.org/10.2307/1969529

[5] F. Caruso, M. C. Ceparano, e J. Morgan "An Inverse-Adjusted Best Response Algorithm for Nash Equilibria", *SIAM Journal on Optimization*, vol. 30, no. 2, pp. 1638–1663, 2020. https://doi.org/10.1137/18M1213701

[6] S. Arenales e A. Darezzo, "Cálculo Numérico - Aprendizagem com apoio de Software", *São Paulo: Cengage Learning*, 2016.

[7] J. M. Martínez e S. A. Santos, "Métodos Computacionais de Otimização", *Rio de Janeiro: Sociedade Brasileira de Matemática*, vol. 20, 1995.

[8] A. H. Sherman,"On Newton-iterative methods for the solution of systems of nonlinear equations"*SIAM Journal on Numerical Analysis*, vol. 15, no. 4, pp. 755–771, 1978. https://doi.org/10.1137/0715050

[9] Y. Yuan, "A trust region algorithm for Nash equilibrium problems", *Pacific Journal of Optimization*, vol. 7, no. 1, pp. 125–138, 2011.

[10] Y. Yuan, "Recent advances in trust region algorithms", *Mathematical Programming*, vol. 151, no. 1, pp. 249–281, 2015. https://doi.org/10.1007/s10107-015-0893-2

[11] Y. Yuan, "A review of trust region algorithms for optimization", *Proceedings of the Fourth International Congress on Industrial and Applied Mathematics (ICIAM)*, Edinburg, 5-9 July 1999, Oxford University Press, Oxford, UK, 271–282, 2000.

[12] A. Dixit, S. Skeath e D. McAdams . G. Garbagio e B. Sartini, *Games of Strategy:Third Edition*, *WW Norton & Compan*, 2009.

[13] H. Bortolossi, G. Garbagio e B. Sartini, *Uma Introdução à Teoria Econômica dos Jogos*, *26º Colóquio Brasileiro de Matemática*, 2007.

BREVE BIOGRAFIA

Luís Felipe Bueno https://orcid.org/0000-0002-8820-6606

Professor associado no Departamento de Ciência e Tecnologia da Universidade Federal de São Paulo em São José dos Campos. É vice coordenador do Programa de Pós Graduação em Matemática Aplicada. Tem trabalhado com pesquisa operacional, com foco no desenvolvimento de métodos computacionais de otimização e aplicações.

Amanda Gomes Vetorazzi https://orcid.org/0000-0002-5627-8138

Graduada em Matemática Aplicada a Negócios pela Universidade de São Paulo, e atualmente é aluna de mestrado no Programa de Pós-Graduação em Pesquisa Operacional da UNIFESP e ITA, em São José dos Campos.