
PRESENÇA DE CAOS EM AMBIENTES DISCRETOS *CHAOS PRESENCE IN DISCRETE ENVIRONMENTS*

SOUZA, Lucas Filipe de Andrade¹; PESCARINI, Lucas²;

¹Graduando do Curso de Engenharia de Computação – Universidade São Francisco;

²Graduando do Curso de Engenharia de Computação – Universidade São Francisco;

lucas.souza@mail.usf.edu.br

RESUMO. Caos e imprevisibilidade são campos inerentes da sociedade, economia, biologia, engenharia e filosofia. É observável, também, por qualquer indivíduo, compondo eventos tanto atraentes, como fractais e complexidade em cenários, quanto problemáticos, como imprevisibilidade na economia. O presente estudo objetiva observar e interpretar melhor a formação destes comportamentos complexos, nomeadamente ordem, organização e caos, a partir de rotinas e definições consideradas básicas no teor da programação. Para tal objetivo, foram empregados códigos para simulações que apresentam alguma sensibilidade a variáveis para cada geração das mesmas, levando em consideração algumas proposições sobre a natureza do caos, notadamente como eventos simples podem apresentar comportamentos caóticos complexos. Ao término do estudo, foi constatada a presença de caos, imprevisibilidade, e de janelas de ordem, relacionando a formação das partículas simuladas com comportamentos observados na natureza, por imagens e vídeos, e por meio de gráficos, extraindo as posições das partículas por tempo. Foi observada a presença de caos e certa imprevisibilidade na simulação, mesmo que com códigos de nível básico, fomentando proposições sobre a teoria e tornando esta mais tangente para futuros desenvolvimentos.

ABSTRACT. Chaos and unpredictability are inherent fields of society, economics, biology, engineering and philosophy. It is also observable by any individual, composing both attractive events, such as fractals and complexity in scenarios, as problematic such as unpredictability in the economy. This study aims to better observe and interpret the formation of these complex behaviors, namely order, organization and chaos, based on routines and definitions considered basic in the programming topic. For this purpose, codes were used for simulations that present some sensitivity to variables for each generation, taking into consideration some propositions about the nature of chaos, notably how simple events can present complex chaotic behaviors. At the end of the study, the presence of chaos, unpredictability, and order windows were found, relating the formation of simulated particles with behaviors observed in nature, through images and videos, and through graphics, extracting the positions of particles by time. Chaos and unpredictability were observed in the simulation, even with basic level codes, fostering propositions about the theory and making it more tangent for future developments.

INTRODUÇÃO

Em computação, Inteligência Artificial (IA) é dada pela habilidade de um sistema aprender com base em dados externos, interpretados pelo mesmo, e utilizar o aprendizado adquirido com estes dados de forma versátil, adaptando-se a tarefas para alcançar objetivos específicos. (KAPLAN e HAENLEIN, 2018).

A pergunta inicial que é feita é, onde é aplicável a teoria do caos em inteligência artificial? Uma das vertentes estudadas em IA é a computação natural, e acredita-se que nela existe a resposta. Uma ideia básica que se encaixa no contexto da computação natural é a de sistema complexo; um sistema complexo é um sistema com um grande número de componentes em interação cuja atividade agregada é não linear (não derivável somando o comportamento dos componentes individuais) e exibindo tipicamente a auto-organização (Morowitz e Singer, 1995; Holland, 1995; Gallagher e Appenzeller, 1999; Rocha, 1999); em um sistema complexo não estamos aptos a entender e explicar seu comportamento analisando apenas seus componentes separadamente (de Castro, Leandro Nunes, 2007).

Lewin 1992, Waldrop 1992 define complexidade como um tipo de comportamento dinâmico que nunca encontra um equilíbrio no qual muitas partículas independentes interagem perpetuamente e procuram acomodação mútua em qualquer uma das várias possibilidades existentes, essas partículas organizam e se reorganizam em estruturas cada vez maiores e evoluídas ao longo do tempo. Como veremos adiante, um comportamento que encontramos ao estudar o caos é a auto-organização.

Frente aos diversos métodos que condizem com o conceito de Inteligência Artificial, o planejamento dita a designação mais favorável para utilização em sistemas de previsão, definindo que os agentes inteligentes devem ser capazes de estabelecer e alcançar seus objetivos, com métodos para representação do estado do universo e como mudanças realizadas irão alterá-lo, a fim de estabelecer a melhor escolha disponível dentre as disponíveis.

Na computação, agente é um programa capaz de realizar tarefas no lugar do usuário. Estes agentes possuem algumas propriedades como inteligência, autonomia e a habilidade de colaboração (JONES; BARTLETT, 2004). Um agente interage com o ambiente através de sensores e atuadores, assim como os humanos possuem olhos, boca e ouvidos como sensores e braços como atuadores, os agentes recebem como entrada dos sensores dados e o atuador escreve estes dados em arquivos, ou outros mecanismos de saída (RUSSEL; NORVIG, 2003).

O estudo do caos se torna interessante ao notarmos que é possível encontrar exemplos do caos em epidemias, populações, mudanças econômicas etc. Por este motivo ele é estudado em diferentes áreas acadêmicas, na física, matemática, química, biologia, economia e em vários outros campos, isto torna o caos um tema interdisciplinar. Na teoria, basicamente tudo que ocorre em função do tempo pode ser considerado caótico (WILLIAMS, G., 1997).

Conjuntamente com computação natural, na qual uma das principais classes de métodos foca em sintetização de fenômenos e eventos naturais, estes podem ser utilizados para demonstração de comportamentos, que incluem cooperação e competição entre diversos agentes em um ambiente dinâmico. (ROZENBERG; BACK; KOK, 2012) (RUSSEL e NORVIG, 2003).

O caos é o lado irregular da natureza, o lado descontínuo e errático. O caos parece estar em todo lugar, desde uma bandeira que se agita ao vento até uma torneira pingando que passa de um padrão estável para um aleatório. O caos aparece no comportamento do clima, de uma

aeronave em voo, não importa qual seja o meio, o caos sempre parece estar lá (GLEICK, J., 1987).

Como exemplo, idealiza-se uma bola com um lado preto e outro branco. Esta bola é colocada em movimento e então observada para saber qual lado está para cima quando a bola parar. Ao realizar este experimento arrisca-se observar que os resultados parecem aleatórios, o motivo para isso é a aleatoriedade na velocidade inicial com que a bola é jogada, ou seja, não se faz possível ter controle preciso sobre a velocidade com que a bola é jogada para assim ser determinado qual lado ficará para cima no final e, com as sucessivas tentativas isto irá aparentar um comportamento aleatório. Esta alteração na velocidade inicial desencadeando um resultado final diferente a cada nova jogada ficou conhecido na literatura como efeito borboleta. Na natureza, uma das coisas mais comuns que se nota é a aparente aleatoriedade. Existem inúmeros tipos de sistemas que exibem aleatoriedade como o crescimento de cristais e o rompimento de objetos sólidos que é o modo mais comum de observamos aleatoriedade no nosso dia-a-dia (WOLFRAM, S., 2002) (WILLIAMS, G., 1997).

Ao contrário do que parece, o Caos não é apenas desordem. Por exemplo, o sucessivo aumento em uma variável de controle de uma equação não geraria necessariamente um aumento nos níveis de caos, porém este aumento pode levar a algum tipo de regularidade, ou seja, ordem.

Observa-se uma equação em função do tempo, temos que alguns valores passam a ser visitados com mais frequência, esses intervalos de ordem são conhecidos como janelas, ou seja, regiões de Periodicidade. São zonas nas quais as trajetórias visitam com mais frequência, ou seja, são intervalos de valores mais visitados em uma função. Grossman & Thomae (1977) descreve o caos como um estado misto onde o desenvolvimento periódico e caótico do tempo se mistura (WILLIAMS, G., 1997).

Se olharmos como exemplo a *logistic equation*, a maior parte das características do caos dependem apenas do parâmetro de controle, para um determinado valor desse parâmetro, a ordem se desenvolve espontaneamente, ou seja, sem causa externa, este processo é denominado auto-organização (WILLIAMS, G., 1997).

G. Wiliams define Auto-organização como o ato pelo qual um sistema de auto propagação, sem influência externa, sai de uma aparente irregularidade para um estado de ordem, ou seja, a partir de um ponto inicial, a ordem se desenvolve espontaneamente sem causa externa. Basicamente, o caos consiste de vários tipos de ordem, camuflados como comportamento aleatório. Sendo assim, ordem dentro do caos é a regra e não a exceção. Podemos encontrar vários exemplos de auto-organização na natureza, desde a organização de um bando de pássaros a moléculas de água se agrupando no chão. (WOLFRAM, S., 2002) (WILLIAMS, G., 1997).

Auto-organização é a característica principal de um comportamento que podemos chamar complexidade. Complexidade é um tipo de comportamento que nunca alcança equilíbrio e as unidades independentes interagem entre si e procuram acomodação de muitas maneiras possíveis, um comportamento complexo tem alguns ingredientes, que serão analisados neste trabalho, como um grande número de partículas independentes, auto-organização e regras locais que governam cada partícula (WOLFRAM, S., 2002) (WILLIAMS, G., 1997).

PORQUE ESTUDAR O CAOS?

Estudar o caos faz perceber que uma simples equação determinística pode criar uma série temporal irregular, o que força a repensar sobre a separação que havia entre determinismo e aleatoriedade. O conceito de aleatório era algo desorganizado, com aparente falta de padrão, mas o problema com este conceito é que existem uma lista de eventos aleatórios que demonstram um padrão, porém ao serem observados isoladamente não podem ser previstos (WILLIAMS, G., 1997).

Programas baseados em regras simples, nem sempre produzem comportamentos simples. O caos nos mostra que não precisamos necessariamente isolar cada componente de um sistema e analisar separadamente cada um deles para entendê-lo, ao invés disso, se olharmos para o sistema como um todo podemos obter avanços maiores, ou seja, se observarmos um evento que foge do padrão, como por exemplo analisar um evento climático apenas com base em um ocorrido não conseguiremos entendê-lo, mas se olharmos ao longo da história por registros parecidos conseguiremos compreendê-lo melhor. (WOLFRAM, S., 2002) (WILLIAMS, G., 1997).

É também observável com clareza o avanço da tecnologia de virtualização, sintetização e modelagem presente desde a indústria de entretenimento, com filmes, jogos e arte, até a indústria de carros, com simulação computadorizada de modelos e aerodinâmica pré-produção, com grande influência destes métodos. (YIYAO e WEI, 2017) (MATEAS, 2002).

Com base nestas observações, no grande impacto social e aplicações destes métodos, surgiu a aspiração de entendimento dos mesmos, focando na sintetização de fenômenos naturais complexos, com realismo, no ambiente virtual, levando em consideração a interação de múltiplos agentes no universo, agrupado a regras físicas em um ambiente dinâmico e caótico. Este trabalho, então, tem por objetivo estudar os métodos utilizados na simulação de tais eventos, propor e aplicar uma combinação destes para sintetizar movimentos naturais em um ambiente limitado, mas complexo.

NATURAL COMPUTING (COMPUTAÇÃO NATURAL)

A fim de estudar movimentos e comportamentos naturais, o tópico da computação natural se destaca. É uma terminologia aplicada a três classes de métodos: inspirados na natureza para técnicas de resolução de problemas, baseados no uso de computadores para sintetizar fenômenos naturais e aqueles que computam materiais naturais, como moléculas, e suas possíveis interações.

Os mais clássicos modelos inspirados na natureza são os de autômatos celulares, computação neural e computação evolutiva.

Estes paradigmas são extraídos de fenômenos naturais diversos, como auto replicação, funcionamento do cérebro e seus neurônios, sistema imunológico, evolução darwiniana, comportamento de grupo. Focando no estudo destes fenômenos na natureza como sistemas de processamento de informação (BRABAZON; O'NEILL; MCGARRAGHY, 2015).

Um exemplo de computação natural é o estudo dos enxames de partículas, da sigla em inglês PS (*Particle Swarm*), um método computacional que otimiza um problema ao iterativamente melhorar o candidato à solução levando em consideração as regras dadas. Foi inicialmente desenvolvido para simulação de comportamento social. A resolução ocorre de forma que as partículas candidatas à solução são movidas no espaço de busca seguindo uma

fórmula matemática sobre a posição e velocidade das mesmas (equação 1). Cada partícula então leva em consideração o melhor resultado dos enxames para a próxima iteração.

PSO é inicializado com um grupo de partículas, dadas como soluções aleatórias, e a partir daí, trabalha para encontrar a melhor solução, com atualizações a cada geração (Figura 1). Essas atualizações seguem, basicamente, dois valores principais: local (melhor valor que a partícula alcançou) e global (melhor valor entre todas as partículas) (Figura 2).

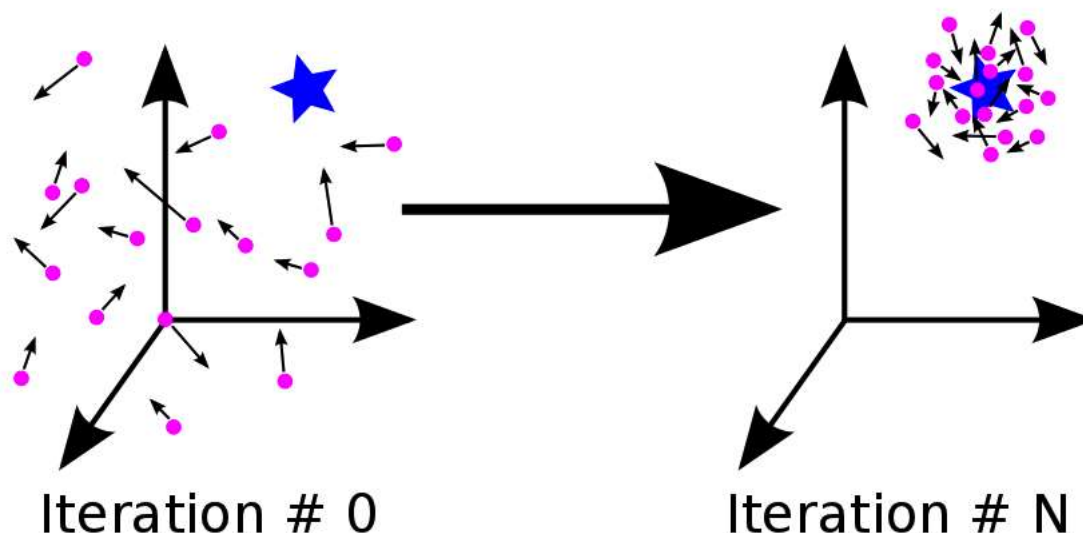


Figura 1 - Comportamento das partículas ao longo das iterações em direção ao objetivo, em um algoritmo PS. (Fonte: *Particle Swarm Optimization Based Source Seeking*—Rui Zou, Vijay Kalivarapu, Eliot Winer, James Oliver, Sourabh Bhattacharya (2017)).

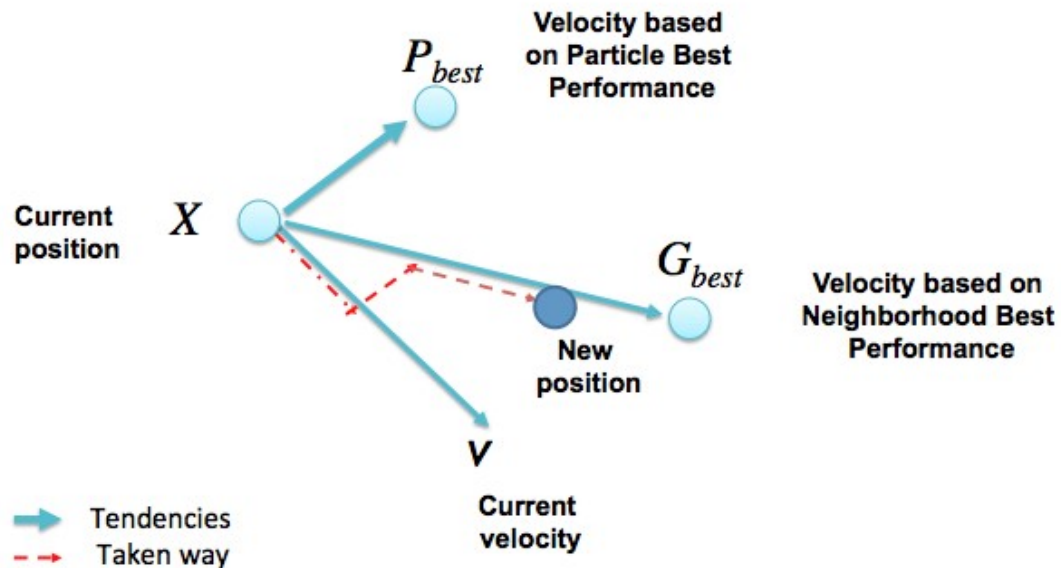


Figura 2 - Como a próxima posição é calculada, levando em consideração velocidade, e valores sociais (global best) e locais (personal best). Vetores resultantes. (Fonte: Chowdhury, S., Tong, W., Messac, A. and Zhang, J., *A Mixed-Discrete Particle Swarm Optimization Algorithm with Explicit Diversity-Preservation, Structural and Multidisciplinary Optimization*, Vol. 47, Issue 3, 2013, pp. 367-388).

Após definir estes valores na iteração atual, a atualização é realizada seguindo as fórmulas:

$$v[i] = v[i] + c1 * rand() * (Ptop[i] - atual[i]) + c2 * rand() * (Gtop[i] - atual[i]); atual[i] = atual[i] + v[i]$$

Equação 1 – Base para o cálculo do novo vetor velocidade da partícula (distância percorrida)

Onde v é dado como velocidade, atual é a solução na qual a partícula se encontra. $Ptop$ e $Gtop$ são as melhores, local e global respectivamente. $Rand$ é um número aleatório de alcance arbitrário. $C1$ e $C2$ são fatores observados durante as iterações, onde $C1$ trabalha a favor da variável local e $C2$ da global (ZOU; et al., 2004).

METODOLOGIA

Dado conhecimento e estudos sobre os tópicos citados, e de comportamentos e interações de partículas na natureza, que a princípio são vistas como caóticas e aleatórias, julga-se que estas podem ser simplificadas e implementadas artificialmente. Foram analisados, discutidos e implementados algoritmos baseados em *Particle Swarm*, e de movimentos erráticos, que focam na reprodução de comportamentos de natureza aleatória, da forma mais próxima ao real possível. Estes foram implementados em DX11 HLSL e C#.

Foram implementados códigos em VB e Java, em ambiente 2D, com código PSO base, que apresentaram resultados não satisfatórios antes mesmo da fase de testes e aquisição de dados para análise. Movimentos erráticos, em loop, e imobilidade foram observados durante depuração. As combinações mais estáveis foram de PSO com regras de evasão, e para o ponto objetivo das partículas os movimentos *brownianos* (*keijiro's ProceduralMotionTrack library*), e cíclico de patrulha (loop com pontos pré-determinados).

Todos os testes foram realizados em uma máquina com sistema operacional Windows 10 build 1903, 16GB RAM DDR4 a 3200MHz, processador Intel i7-8750H com 9M cache a 3.00GHz, GeForce 2080 Max-Q 8GB GDDR6 a 6GHz.

Realizada coleta de dados durante testes, alimentando um arquivo de texto com coordenadas, conforme Código 1, e produzindo capturas de tela.

```
//contador de posicao para graficos
    contadordeframes++;
    if (contadordeframes % 300 == 0)
    {
        contadordesegundos++;
        boidComputeBuffer.GetData(this.boidsData);
        for (int j = 0; j < BoidsCount; j++)

System.IO.File.AppendAllText(@"C:\Users\USER\Desktop\BOIDS.txt", "BOID NUMERO " + j + " POS : " +
this.boidsData[j].Position.ToString() + " [Segundo] " +
contadordesegundos + Environment.NewLine);
    }
//end contador
```

Código 1 – Coleta das posições de cada partícula, em arquivo de texto.

Foi utilizada como métrica de comparação de dados, pesquisas sobre comportamento natural de partículas ou objetos/animais e seu comportamento inteligente de patrulha e caminhos seguidos com caminhos formados e patrulhas resultantes do código aplicado. Gráficos construídos e alimentados a partir da localização das partículas no espaço em determinado instante, assim como densidade de partículas, por tempo e área. A comparação de resultados de diferentes interações e iterações do código, servirá para averiguação da aleatoriedade e naturalidade alcançada. A média de iterações é de 40 a 60 por segundo, sendo que os dados serão colhidos como triplas ordenadas (x,y,z), utilizando módulo destas em função do tempo ou *frame* (t) para cada partícula no espaço, desconsiderando os obstáculos móveis.

Meta-análise e passos do estudo:

- Observação do comportamento natural;

Registro de dados que determinam os comportamentos, na natureza, a serem simulados, a um nível visual.

- Definir hipóteses;

Definido objetivo, avaliar se o mesmo será capaz de cumprir com o desejado.

- Inserção dos dados da observação, principalmente rotinas, em algoritmos e objetos;

Com os dados observados, definir as regras a serem utilizadas nos algoritmos. Foi aplicado o código padrão *Particle Swarm*, com regras de dispersão ao lidar com evasão de obstáculos.

- Alterar as variáveis nas iterações, de forma a testar interação entre os mesmos;

Realizar o experimento, com parâmetros alterados, para validação das hipóteses.

- Observar resultados;

Analisar os resultados esperados nas hipóteses, com os resultados obtidos nas iterações, gerando gráficos e comparações como localização x tempo.

- Repetir experimento;

Replicar o experimento, para concretização dos resultados.

- Conclusões sobre o experimento

Com base nas hipóteses, testes e experimentos, analisar e determinar os resultados deste estudo.

PARTICLE SWARM, UM SISTEMA COMPLEXO QUE SIMULA O FLOCK

Partindo para aplicação da ideia, o foco é a geração de um sistema que alcance caos e comportamentos complexos, e visualmente represente uma cena natural. Por exemplo, um bando de pássaros. Para G. Willians, um comportamento complexo possui ao menos seis características, grande número de partículas, agentes ou componentes semelhantes, mas independentes; Dinamismo, cada agente atua continuamente e responde aos vizinhos de maneiras novas; Adaptabilidade, o sistema se ajusta a novas situações. Auto-organização, a ordem se forma espontaneamente; Regras locais que governam cada agente; Progressão hierárquica, as regras se tornam mais eficientes e a estrutura mais complexa.

No *particle swarm* nota-se que este possui todas estas características, sendo assim, está presente no *particle swarm* uma característica extremamente intrínseca ao caos e a sistemas complexos originados dele, a auto-organização, sempre que uma perturbação é inserida no sistema as partículas são dispersas, o que é o caos sendo gerado, porém assim que essa perturbação termina as partículas tendem à aglomeração, ou seja, auto-organização.

O *particle swarm* se baseia em regras simples que fazem com que as partículas alcancem um determinado resultado, porém nota-se uma certa complexidade em seu comportamento, assim como Stephen Wolfram nos mostra, na maioria das vezes regras notavelmente simples dão origem a comportamentos complexos. Dado isto, julga-se possível aplicar essas regras para simular comportamentos naturais, como visto no código 2, é criado um modelo de partícula que pretende representar aves, estas possuem os parâmetros de controle, coordenada, direção, perímetro para evitar colisão de partículas e se é predador ou não.

```
struct Boid
{
```



```
float3 Posicao;  
float3 Direcao;  
int Predador;  
float perimetro;  
  
...  
};
```

Código 2 – Estruturação dos boids

Os agentes possuem regras básicas, os que não são predadores precisam manter um perímetro de distância entre eles, devem ir em direção ao sol e precisam fugir de predadores que venham a aparecer. Os agentes que representam predadores por outro lado possuem outro conjunto de regras básicas, eles devem ir em direção ao grupo de pássaros e não pode entrar no perímetro de outro predador. Ao definir as variáveis neste ambiente, manifesta-se o fator de caos, que neste caso se torna o predador.

No código 3 se apresenta a codificação das regras para os agentes, os valores atualizados são utilizados nos cálculos da equação 1.

```
dist = distance(boid.Posicao, outroBoid.Posicao);  
if (dist <= perimetroSeparacao) {  
    float3 diff = boid.Posicao - outroBoid.Posicao;  
    float diffTamanho = length(diff);  
    float scaler = clamp(1.0 - diffTamanho /  
perimetroSeparacao, 0.0, 1.0);  
    separacao += diff * (scaler / diffTamanho);  
    ContadorSeparacao += 1;  
}  
if (dist <= RaioCoesao){  
    coesao += outroBoid.Posicao;  
    contadorCoesao += 1;  
}  
if (dist <= RaioAlinhamento){  
    alinhamento += outroBoid.Direcao;  
    contadorAlinhamento += 1;  
}  
if (outroBoid.Predador == 1){  
    if (dist <= raioDeFuga){  
        boid.State = 2;  
        float3 revDirecao = boid.Posicao -  
outroBoid.Posicao;
```

```

        alinhamento = outroBoid.Direcao + float3(10,
10, 10);

        coesao = revDirecao;
        separacao += revDirecao * length(revDirecao);
        contadorAlinhamento = contadorCoesao =
ContadorSeparacao = 2;
        speed *= 3;
        break;
    }
}

```

Código 3 – Aplicação das regras e atualização de valores para cada agente.

Ao definir as regras para os agentes pretende-se reproduzir movimentos com naturalidade, dando a impressão de que os agentes são de fato um grupo de pássaros real. Na natureza existe um comportamento estudado por ecologistas e biólogos denominado *flock*, é uma associação de aves onde os participantes buscam um ao outro, ativamente iniciando e mantendo sua associação, estas aves podem convergir em comida, água e habitat, além de se defenderem contra predadores (Morse 1970). Quando o *particle swarm* é executado, observa-se que seu comportamento se assemelha ao *flock*, como comparável na figura 3 e figura 4. Pode se notar na figura 4 um comportamento aparentemente natural, o esperado visto que o objetivo da área de computação natural é imitar comportamentos da natureza, além disso é possível observar uma aparente aleatoriedade do seu comportamento ao longo do tempo.



Figura 3 - *Flock* de pássaros criando formas no céu. (Fonte: <https://i.ytimg.com/vi/bb9ZTbYGRdc/maxresdefault.jpg>, acesso em 16/11/2019).



Figura 4 - Simulação com *particle swarm* realizada no *Unity3D*. (Fonte: Próprio autor).

TEORIA DO CAOS OBSERVÁVEL AO SIMULAR O FLOCK

Como descrito no início deste trabalho a teoria do caos se baseia no caos e na ordem, na transição entre estes estados, ao inserirmos um predador no ambiente do *particle swarm*, é gerado caos no sistema fazendo com que as partículas se espalhem sempre que se aproxima da posição em que elas se encontram, assim que o predador se afasta as partículas tendem a se reagrupar sem que haja qualquer interferência externa, este comportamento de auto organização é originado da ordem, assim como descrito anteriormente neste trabalho, esta transição observável de caos gerado pelo predador e auto organização gerado pelos pássaros é um exemplo da teoria do caos presente no sistema.

Tomando como ponto inicial a figura 5, nela nota-se que o sistema está com suas partículas em estado de desordem pois a posição inicial de cada partícula é aleatório. Seguindo para a figura 6 constata-se que as partículas estão em processo de auto-organização, portanto registra-se a partida de um caos inicial e o sistema caminha para a ordem, sendo nítido que se encontram mais agrupadas neste momento.



Figura 5 – Partículas ordenadas em grupos para o objetivo (Fonte: Própria).

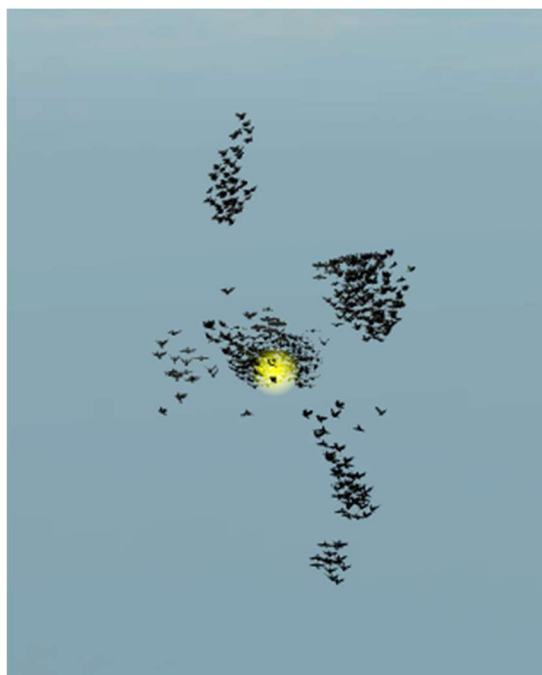


Figura 6 – Dissipação (Fonte: Própria).

Nota-se na figura 7 um estado de ordem, após o de auto organização apresentado na figura 6, as partículas se encontram sem perturbação, mantendo-se agrupadas e organizadas, neste cenário pode se observar que o predador encontra-se longe da trajetória delas, portanto não causa alteração no comportamento das mesmas, por ser o fator que suscita o caos no ambiente assim que o predador se aproxima da trajetória de um dos pássaros, é aparente que o grupo tende a se dissipar, como observado na figura 8.

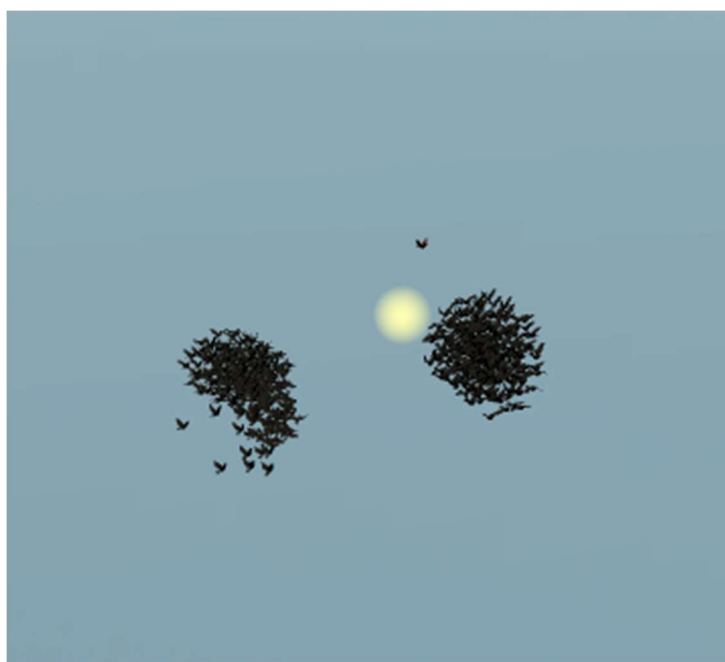


Figura 7 - Agrupamento (Fonte: Própria).



Figura 8 – Dissipação de um dos grupos que sofreu ataque (Fonte: Própria).

Por final, se dá um ciclo dentro da teoria do caos, tendo início com o ambiente caótico, figura 5, passando pela ordem e auto-organização na figura 6 e figura 7, voltando a demonstrar caos no ambiente na figura 8 e enfim ocorre auto-organização e ordem novamente na figura 9 e figura 10. Tendo que o caos é essa alternância de ordem e desordem, é dada como perceptível sua presença ao simularmos o *flock*.



Figura 9 – Fim da dissipação do grupo e início da reorganização (Fonte: Própria).



Figura 10 – Reorganização do grupo atacado (Fonte: Própria).

Por se tratar de um comportamento que ocorre em função do tempo, é observável na figura 11 a atuação de todas as partículas do sistema ao longo de 17 minutos, nesse gráfico fica evidente um padrão, porém com picos que se destacam, é perceptível que existe ruído na função, o que faz com que o gráfico não seja completamente homogêneo. Comparando a figura 12, que não possui um predador no ambiente, nota-se que existe uma homogeneidade, não havendo o mesmo ruído que na figura 11.

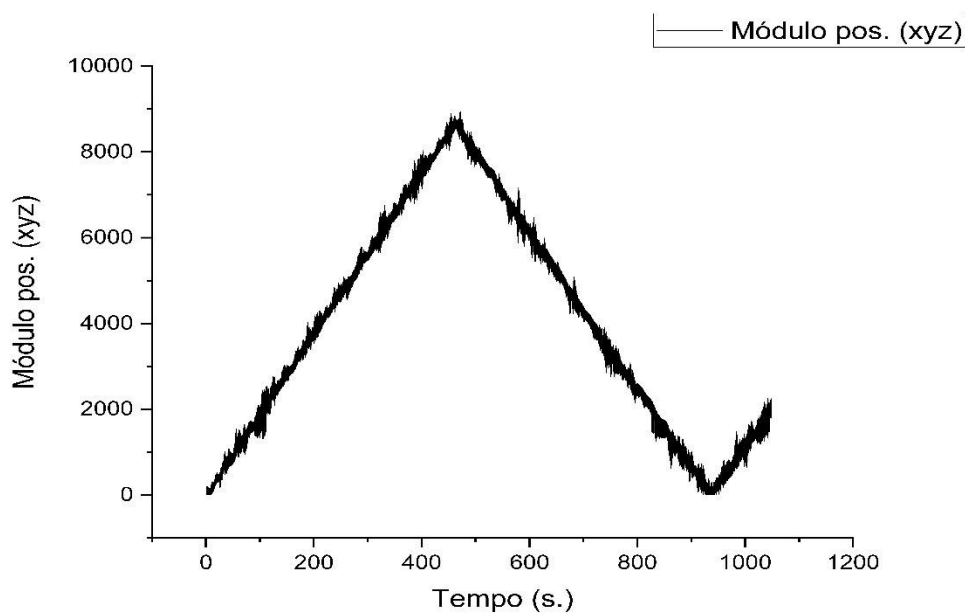


Figura 11 – Gráfico posição x tempo de todas (1000) as partículas do sistema (Fonte: Própria).

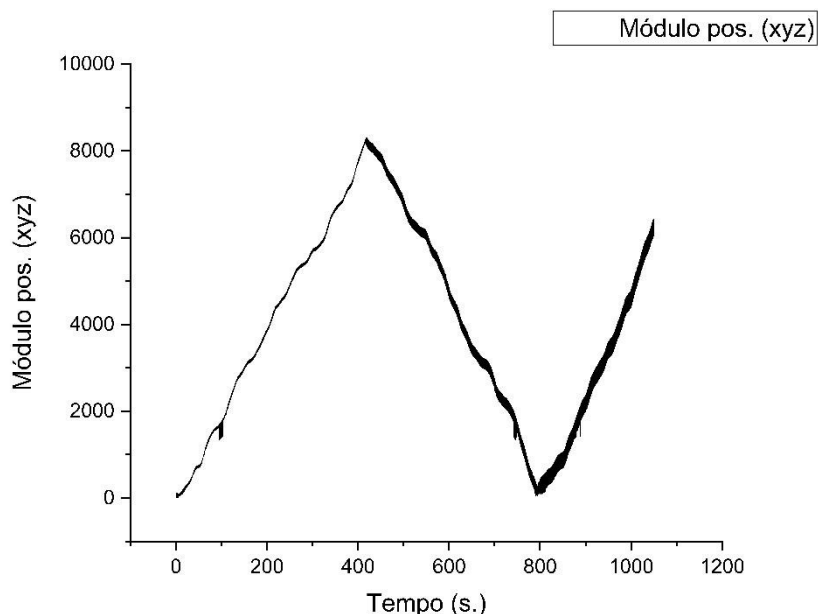


Figura 12 – Gráfico posição x tempo de todas (1000) as partículas do sistema. Ambiente sem predador (Fonte: Própria).

Isolando partículas aleatórias e analisando seus módulos da trajetória, figura 13, é perceptível, em detalhe, tanto se comparadas, como se analisadas individualmente, o ruído e instabilidade, assim como janelas de ordem dispersas. Ao comparar com o gráfico da figura 14 onde o ambiente não possui predador nota-se que as três partículas se comportam de maneira homogênea. A figura 15 e figura 16 demonstram este mesmo comportamento em uma janela menor de tempo, onde é observável com mais precisão o ruído que ocorre no ambiente com o predador e como a função se comporta no ambiente sem predador.

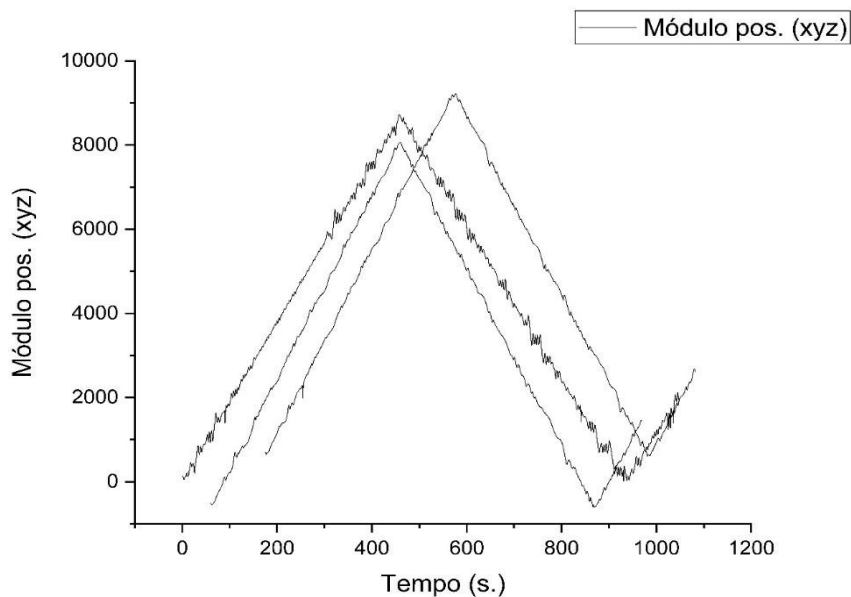


Figura 13 – Gráfico isolado de 3 partículas diferentes aleatórias. Ambiente com predador. (Fonte: Própria).

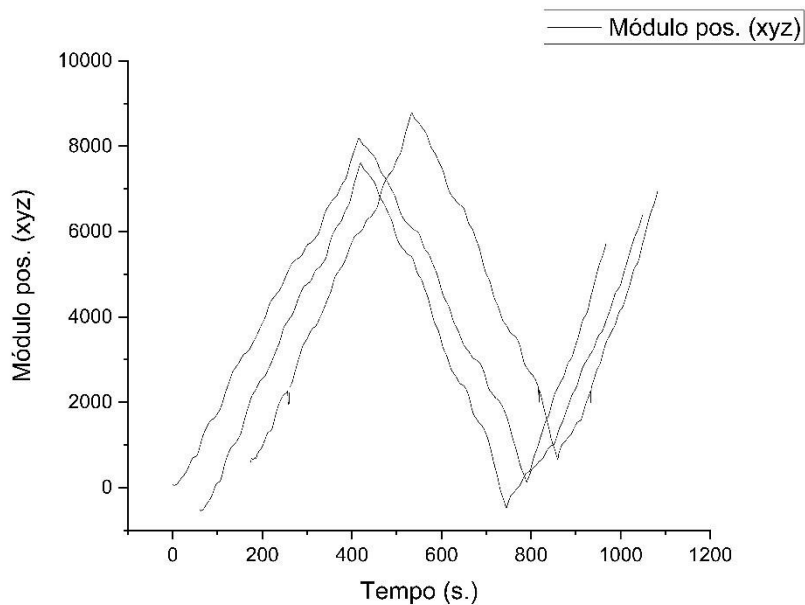


Figura 14 - Gráfico isolado de 3 partículas diferentes aleatórias. Ambiente sem predador.
(Fonte: Própria).

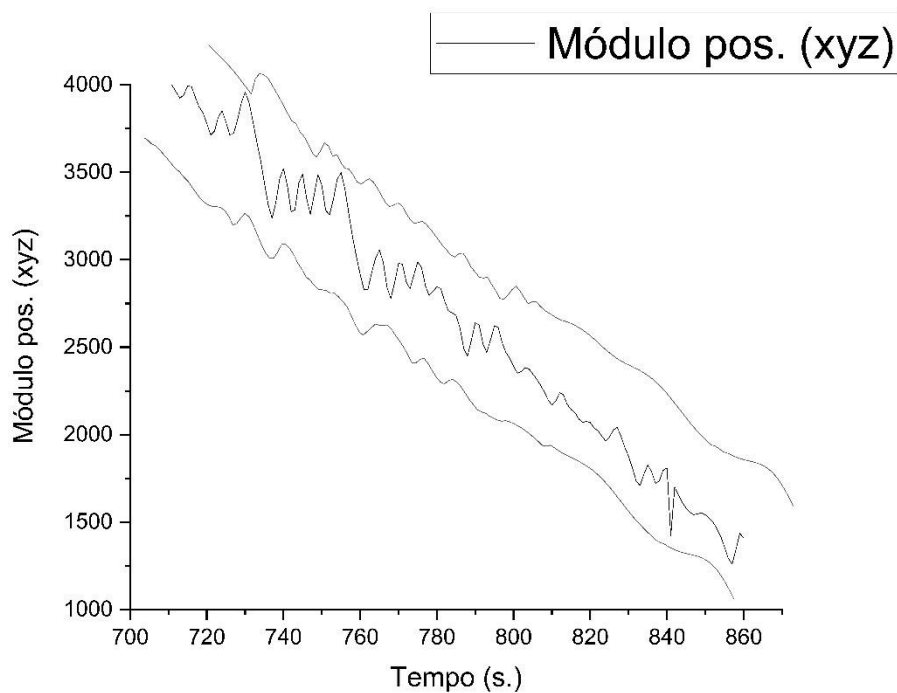


Figura 15 – Gráfico isolado de 3 partículas diferentes aleatórias, em menor janela de tempo.
Ambiente com predador (Fonte: Própria).

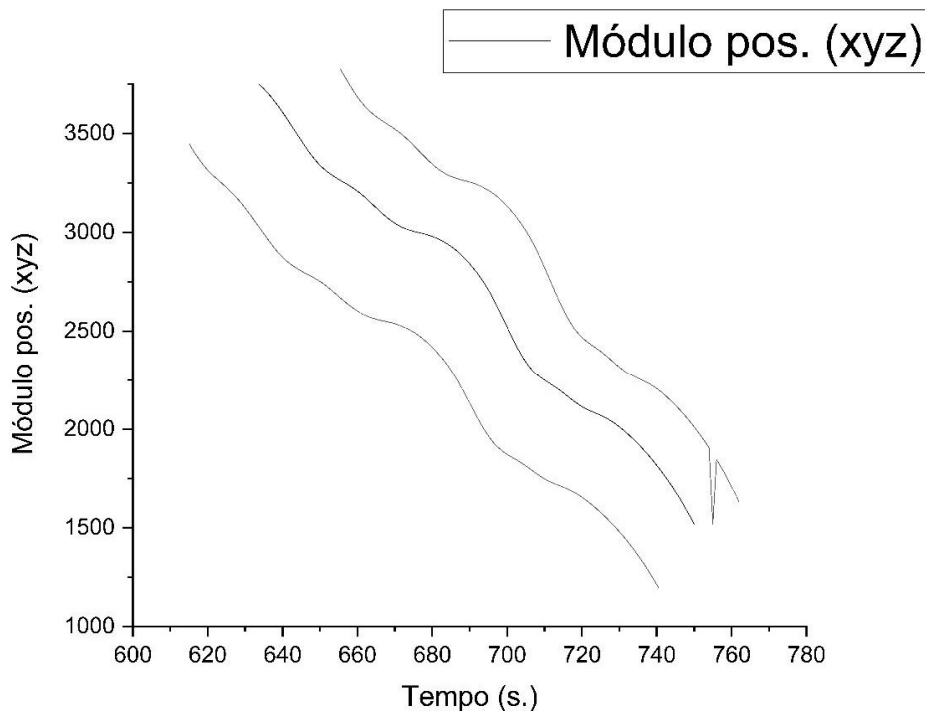


Figura 16 – Gráfico isolado de 3 partículas diferentes aleatórias, em menor janela de tempo. Ambiente sem predador (Fonte: Própria).

CONCLUSÃO

Técnicas para prover realismo em simulações são importantes em diversas áreas, como automobilística e aeroespacial. Como apresentado neste artigo, existe uma área específica de inteligência artificial que busca imitar comportamentos naturais. Ao ser aplicada uma técnica oriunda dessa área, conseguimos obter resultados satisfatórios na simulação de um determinado evento encontrado na natureza, o *flock*. Ao utilizar o *Particle Swarm*, a simulação apresenta um nível de naturalidade elevado, tendo um comportamento semelhante ao observado no mundo real, cumprindo, assim, um dos objetivos deste trabalho, que é obter naturalidade em um evento simulado no computador.

Foi possível observar e analisar todos os elementos envolvidos na teoria do caos, sendo eles o caos e a ordem, e o sistema complexo que emerge dela, sendo o *Particle Swarm* um bom exemplo de sistema complexo. Foi possível observar a auto-organização e afirmações que se provaram verdadeiras, como a de Stephen Wolfram, que diz que regras simples nem sempre produzem comportamentos simples. Sendo assim, ao final deste trabalho, foi obtido um resultado satisfatório, tornando um pouco da natureza do caos e como ele aparece na natureza inteligível, além de compreender melhor como observá-lo em sistemas dinâmicos e proporcionar diversas formas para abordagens no tema, dentro do âmbito da computação.

REFERÊNCIAS

ADAMATZKY, A. **Game of Life Cellular Automata**. Springer. 2010.



BANSOD, N. A.; KULKARNI, M.; PATIL, S.H. **Soft Computing- A Fuzzy Logic Approach**. 2005. 50. Ciência da Computação - Bharati Vidyapeeth College of Engineering, India. 2005.

BRABAZON, A.; O'NEILL, M.; MCGARRAGHY, S. **Natural Computing Algorithms**. Springer Verlag, 2015.

CHATURVEDI, D. K., **Soft Computing: Techniques and Its Applications in Electrical Engineering**. Springer, 2008.

CHOWDHURY, S., TONG, W., MESSAC, A. AND ZHANG, J., **A Mixed-Discrete Particle Swarm Optimization Algorithm with Explicit Diversity-Preservation, Structural and Multidisciplinary Optimization**, Vol. 47, Issue 3, 2013, pp. 367-388

CLARK, Jack - **Why 2015 Was a Breakthrough Year in Artificial Intelligence**. 2015. Disponível em: <<https://www.bloomberg.com/news/articles/2015-12-08/why-2015-was-a-breakthrough-year-in-artificial-intelligence>>. Acesso em: 12/03/2019.

GARDNER, M. **Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life"**. Scientific American. 223, 223, 120–123, Outubro 1970.

GLEICK, J. **Chaos: Making a new Science**. Viking Books, 1987.

KAPLAN ANDREAS; MICHAEL HAENLEIN. **Siri, Siri in my Hand, who's the Fairest in the Land? On the Interpretations, Illustrations and Implications of Artificial Intelligence**. Business Horizons, 62,1,15-25, Janeiro 2019.

KENNEDY, J.; EBERHART, R.C. **Swarm Intelligence**. Morgan Kaufmann. 1995.

MATEAS, M. **Interactive Drama, Art and Artificial Intelligence**. 2002. 284. Ciência da Computação - Carnegie Mellon University, Pittsburgh PA. 2002.

ROZENBERG, G.; BACK, T.; KOK, J., **Handbook of Natural Computing**. Springer Verlag, 2012.

RUSSEL, S. J.; NORVIG, P., **Artificial Intelligence: A Modern Approach**. Segunda edição. Upper Saddle River, New Jersey: Prentice Hall, 2003.

RUSSEL, S. J.; NORVIG, P., **Artificial Intelligence: A Modern Approach**. Terceira edição. Upper Saddle River, New Jersey: Prentice Hall, 2009.

SCHAEFFER, J. **Didn't Samuel Solve That Game? One Jump Ahead**. Springer, Boston, MA, 2009.

SCHIFF, J. L. **Cellular Automata: A Discrete View of the World**. Wiley & Sons, Inc. 2011.



<http://ensaios.usf.edu.br>

TAKAHASHI, Keijiro **Boids**. 2014 <<https://github.com/keijiro/Boids>> Acesso em Julho de 2019.

WILLIAMS, G. **Chaos theory tamed**. Joseph Henry Press, 1997.

WOLFRAM, S. **A New Kind of Science**. Wolfram Media, 2002.

YIYAO, H.; WEI, H. **Aerodynamic Simulation and Optimization Design of Vehicle External Flow Field based on ANSYS**. 2017. 6. Engenharia mecânica e elétrica - Southwest Petroleum University, Chengdu, China. 2017.

ZOU, R.; KALIVARAPU, V.; WINER, E.; OLIVER, J.; BHATTACHARYA, S., **Particle Swarm Optimization-Based Source Seeking** IEEE Transactions on Automation Science and Engineering. 12, 3, 865 - 875, julho 2015.