

Projeto 3

João Vitor Araki Gonçalves (176353) e Tiago Loureiro Chaves (187690)

MC906A - Introdução à Inteligência Artificial - 1s2019

Resumo

Neste projeto propõe-se uma simulação bidimensional do comportamento apresentado por pássaros ao voarem em bando (*flocking behavior*), baseada no modelo clássico desenvolvido por Craig Reynolds em 1987 [1].

O algoritmo apresentado, entretanto, aplica um sistema *fuzzy* para decidir a orientação de voo de cada agente (pássaro simulado – *boid*), ao invés de descrever todo o modelo de direção por relações vetoriais.

O modelo clássico e as variações *fuzzy* sobre ele são descritas na Seção 2, os resultados obtidos são apresentados na Seção 4, e na Seção 5 argumentamos que o uso de regras linguísticas levou a uma simulação superior, após uma análise geral dos dados.

1 Objetivos

Uma vez que queremos simular o comportamento de um animal, é admissível conjecturar que mesmo medidas de valores quantificáveis, como distância relativa, são percebidas de forma vaga e inexata (p.e. "muito perto", "perto", "distante" e "muito distante").

Logo, o uso de lógica fuzzy para representar as percepções consideradas (distância, posição e orientação relativas) e as ações tomadas a partir delas é natural.

Dessa forma, tem-se como intenção que a simulação criada apresente resultados análogos aos alcançáveis por uma modelagem *crisp*, mesmo utilizando regras de inferência fuzzy para o cálculo dos comportamentos de direção de voo: alinhamento, coesão e separação (*steering behaviors*).

2 Problema

Modelar o comportamento apresentado por grupos de pássaros, chamados de *flocks*, ao voar em conjunto como ilustra a Figura 1 (de forma similar a um cardume de peixes, ou a um enxame de abelhas), e então simulá-lo em um plano bidimensional.

Tal comportamento, denominado *flocking* [2], apesar de manifestar coordenação e complexidade, é considerado um comportamento emergente que surge de interações locais entre os pássaros, sem nenhum tipo de organização centralizada ([3] mostra que esturnídeos alinham seus movimentos com apenas as 7 aves mais próximas).

Entretanto, não há uma definição formal de como um coletivo de pássaros deve se comportar para compor um *flock*, portanto, adotamos que ao apresentar as seguintes propriedades um grupo de agentes caracteriza um *flock* [4]:

- Estão próximos um do outro
- Movem-se com aproximadamente a mesma velocidade
- Movem-se aproximadamente na mesma direção
- Agentes que se encontram tendem a formar um *flock*



Figura 1: *Flocking behavior* ¹

O modelo *crisp* clássico, proposto por Reynolds [5], considera 3 regras básicas: alinhamento, coesão e separação. O conjunto dessas regras é chamado de *steering behaviors*, e cada uma é descrita na Seção 2.1.

2.1 Modelo clássico

As representações artificiais de um pássaro (triângulos na simulação apresentada) são denominadas *boids*. Cada *boid* é tratado como um ponto material de massa 1, de forma que podemos simplificar a segunda lei de Newton:

$$\sum \mathbf{F} = m \cdot \mathbf{a} \xrightarrow{m=1} \mathbf{a} = \sum \mathbf{F} \quad (1)$$

assim, a aceleração \mathbf{a} sofrida é igual à força resultante experimentada ($\sum \mathbf{F}$).

Se considerarmos que, a cada instante de tempo, um *boid* percebe o seu ambiente, a partir de suas percepções toma decisões sobre sua nova velocidade e orientação no instante seguinte, e então altera o ambiente ao agir sobre ele, podemos classificar os *boids* como agentes [6], ou, especificamente, como agentes reativos simples [7].

Para modelar a percepção local dos pássaros define-se uma vizinhança \mathcal{N} de um *boid* como sendo formada pelos agentes dentro de um círculo de raio R centrado em sua posição (ou uma esfera, no caso tridimensional). Deste modo, a vizinhança de um *boid* i é o conjunto:

$$\mathcal{N}_i = \{ \text{agentes } j \mid \|\mathbf{p}_j - \mathbf{p}_i\| \leq R, j \neq i \} \quad (2)$$

onde $\|\mathbf{p}_j - \mathbf{p}_i\|$ é a distância entre os agentes i e j , e $\mathbf{p}_i \in \mathbb{R}^2$ é o vetor posição de i .

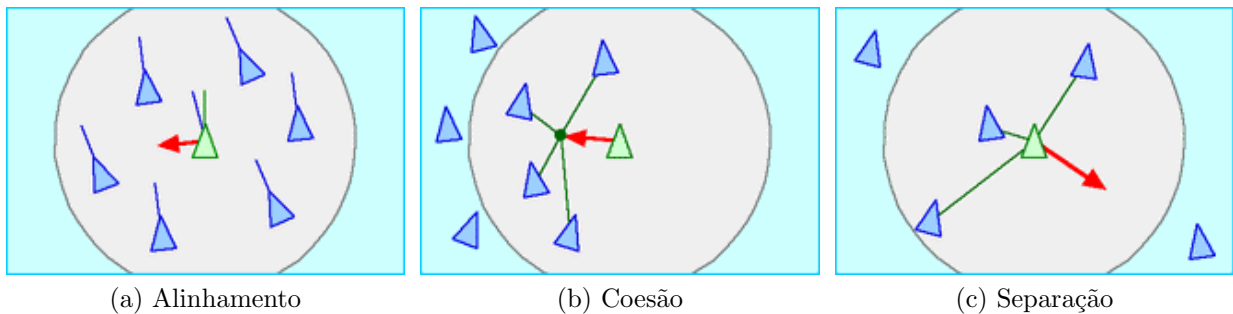


Figura 2: *Steering behaviors*, mostrando as forças referentes a cada regra (em vermelho)

¹[Flight of the Starlings \(National Geographic\)](#), [Watch a Mesmerising Swarm of Starlings \(National Geographic\)](#)

Diferentes *steering behaviors* podem apresentar diferentes raios de percepção, porém todos levam a forças que influenciam mudanças na direção de voo. Como cada agente é definido pela sua posição \mathbf{p} , velocidade \mathbf{v} e aceleração \mathbf{a} , a cada instante de tempo dt (frame da simulação) atualizam-se estes valores, conforme:

$$\begin{cases} \mathbf{a} = \frac{\tilde{\mathbf{a}}}{\|\tilde{\mathbf{a}}\|} \min(\|\tilde{\mathbf{a}}\|, f_M), & \text{onde } \tilde{\mathbf{a}} \stackrel{(1)}{=} \omega_a \mathbf{F}^a + \omega_c \mathbf{F}^c + \omega_s \mathbf{F}^s \\ \mathbf{v} = \frac{\tilde{\mathbf{v}}}{\|\tilde{\mathbf{v}}\|} \min(\|\tilde{\mathbf{v}}\|, v_M), & \text{onde } \tilde{\mathbf{v}} = \mathbf{v} + \mathbf{a}dt \\ \mathbf{p} = \mathbf{p} + \mathbf{v}dt \end{cases} \quad (3)$$

em que f_M e v_M são a força total e a velocidade máximas, ω são pesos, e as forças de direcionamento que agem sobre um agente i são:

Alinhamento: orientar-se para o direcionamento médio dos vizinhos.

$$\mathbf{F}_i^a = \left(\frac{1}{|\mathcal{N}_i^a|} \sum_{j \in \mathcal{N}_i^a} \mathbf{v}_j \right) - \mathbf{v}_i \quad (4)$$

Coesão: orientar-se para o centro de massa dos vizinhos.

$$\mathbf{F}_i^c = \left(\frac{1}{|\mathcal{N}_i^c|} \sum_{j \in \mathcal{N}_i^c} \mathbf{p}_j \right) - \mathbf{p}_i \quad (5)$$

Separação: orientar-se de forma a evitar colisões com os vizinhos.

$$\mathbf{F}_i^s = \frac{1}{|\mathcal{N}_i^s|} \sum_{j \in \mathcal{N}_i^s} \frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_j - \mathbf{p}_i\|^2} \quad (6)$$

2.2 Variáveis de entrada e saída

Como queremos modelar a forma de raciocínio e tomada de decisão de um pássaro, propomos que as forças exercidas sobre um agente devem: (i) representar as intenções de direcionamento que um animal teria (ii) ser obtidas utilizando regras linguísticas, substituindo as equações 4, 5 e 6.

Em vista disso, adotamos como principais fatores que interferem na mudança de direcionamento de um *bird* a distância, a orientação e a posição relativa de outros *birds* (apelidados *flockmates*).

Portanto, temos 3 sistemas de inferência, um para cada *steering behavior*, todos tendo como variável de saída uma mudança de direcionamento (**headingChange**), apresentada na Figura 3, com a qual podemos até representar graficamente as intenções de direção de um agente.

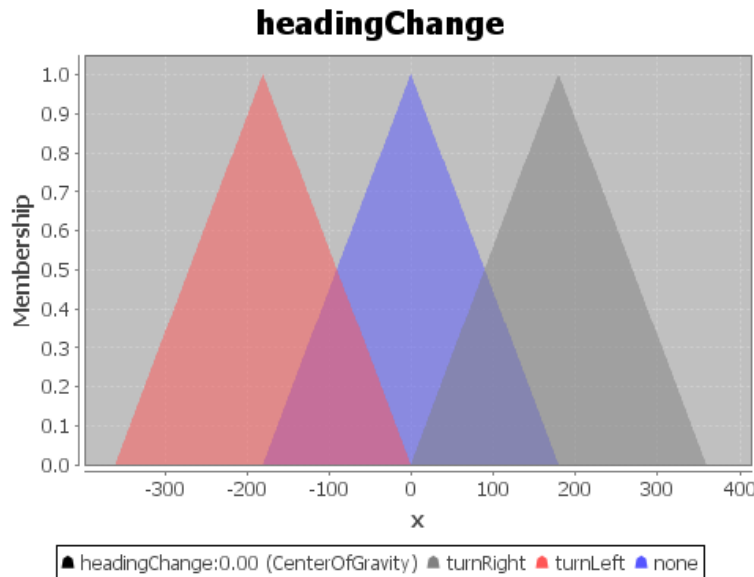


Figura 3: Funções de pertinência da variável de saída, representando a mudança de direção (em graus)

Como a coesão e a separação refletem intenções de mesma direção e sentidos opostos, diferentes raios de percepção foram utilizados ($R^c > R^s$), mas ambas tem como variável de entrada para o sistema de inferência a distância dos agentes vizinhos (em porcentagem do raio de percepção) e sua posição relativa (ou seja, o deslocamento angular do vizinho em relação à atual orientação de voo). As Figuras 4 e 5 mostram essas variáveis linguísticas.

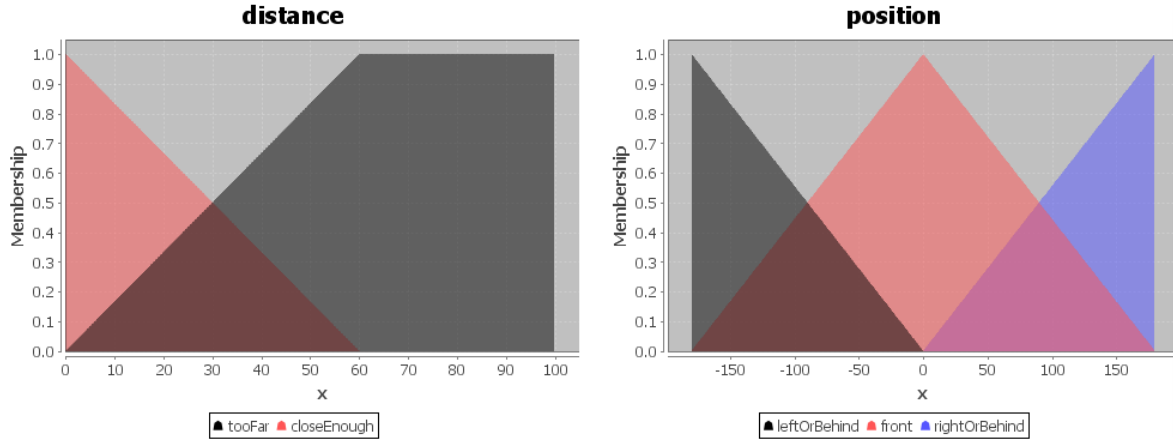


Figura 4: Funções de pertinência das variáveis de entrada para coesão

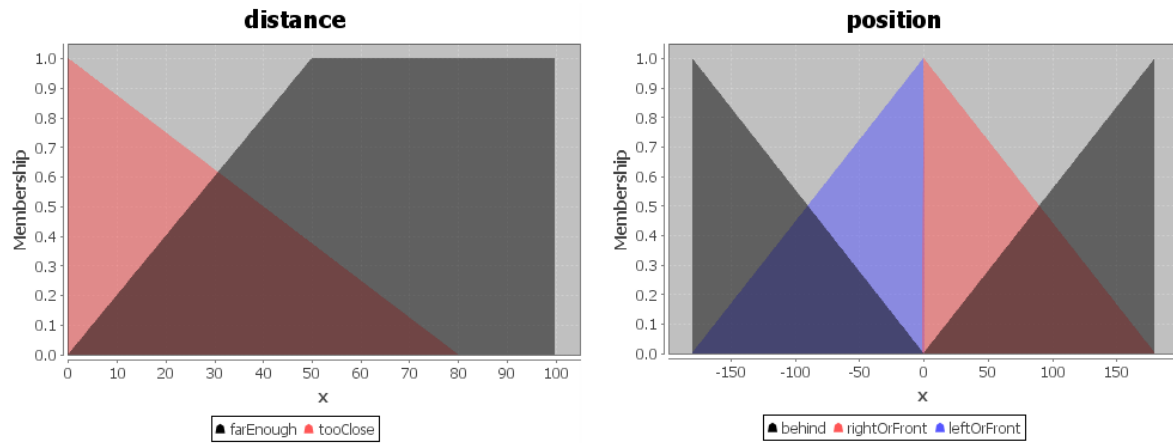


Figura 5: Funções de pertinência das variáveis de entrada para separação

Já a direção de alinhamento é calculada baseada apenas na direção de voo dos *boids* dentro da vizinhança (medida pela diferença angular em relação à atual direção de voo). A Figura 6 mostra a variável de entrada definida.

2.3 Sistema de regras e modelo de inferência adotado

Com as variáveis definidas na Seção 2.2, implementamos as seguintes regras de inferência para determinar os ângulos de direcionamento que caracterizam cada *steering behavior*, pelo método de Mamdani [8]:

Alinhamento:

1. IF direction IS same THEN headingChange IS none;
2. IF direction IS left THEN headingChange IS turnLeft;
3. IF direction IS right THEN headingChange IS turnRight;

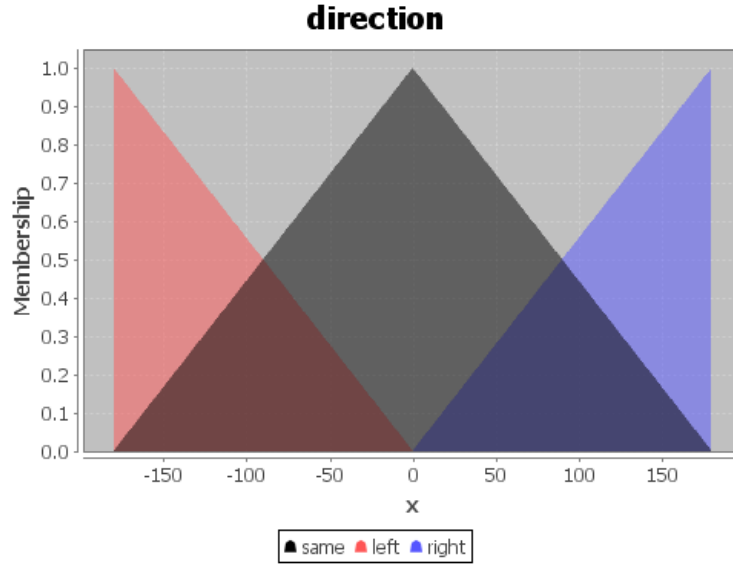


Figura 6: Funções de pertinência da variável de entrada para alinhamento

Coesão:

1. IF distance IS closeEnough THEN headingChange IS none;
2. IF distance IS tooFar AND position IS front THEN headingChange IS none;
3. IF distance IS tooFar AND position IS leftOrBehind THEN headingChange IS turnLeft;
4. IF distance IS tooFar AND position IS rightOrBehind THEN headingChange IS turnRight;

Separação:

1. IF distance IS farEnough THEN headingChange IS none;
2. IF distance IS tooClose AND position IS behind THEN headingChange IS none;
3. IF distance IS tooClose AND position IS leftOrFront THEN headingChange IS turnRight;
4. IF distance IS tooClose AND position IS rightOrFront THEN headingChange IS turnLeft;

Finalmente, como cada variável de saída é um conjunto fuzzy, aplicamos o método do centróide (e inferência max-min) para obter um número *crisp* que representa a mudança de orientação.

2.4 Especificidades e restrições

A cada frame simulado calculamos a nova posição de todos os agentes utilizando os ângulos resultantes da defuzzificação das regras de alinhamento (θ^a), coesão (θ^c) e separação (θ^s).

Para representarmos todas as "intenções" de um *boïd* i incitadas por seus vizinhos, definimos um vetor com a mudança de orientação que cada *boïd* j desperta:

$$\begin{cases} \hat{F}_j^a = (\cos\theta_j^a, \sin\theta_j^a), & \text{para } j \text{ em } \mathcal{N}_i^a \\ \hat{F}_j^c = (\cos\theta_j^c, \sin\theta_j^c), & \text{para } j \text{ em } \mathcal{N}_i^c \\ \hat{F}_j^s = (\cos\theta_j^s, \sin\theta_j^s), & \text{para } j \text{ em } \mathcal{N}_i^s \end{cases} \quad (7)$$

Assim, redefinimos as forças de *steering* de modo que toda a vizinhança percebida (determinada pelos raios R^a , R^c e R^s) influencie na orientação final, e com elas atualizamos os agentes segundo a equação 3.

$$F_i^a = \sum_{j \in \mathcal{N}_i^a} \hat{F}_j^a, \quad F_i^c = \sum_{j \in \mathcal{N}_i^c} \hat{F}_j^c, \quad F_i^s = \sum_{j \in \mathcal{N}_i^s} \frac{\hat{F}_j^s}{\|\mathbf{p}_j - \mathbf{p}_i\|^2} \quad (8)$$

Nota-se que a influência de cada *flockmate* na força de separação manteve-se inversamente proporcional ao quadrado de sua distância. Consideramos isso válido pois a distância $\|\mathbf{p}_j - \mathbf{p}_i\|$ já é calculada de qualquer forma, por ser variável de entrada para as regras de inferência de separação e coesão.

Uma restrição considerável foi observada na simulação ao calcular-se todas as forças de acordo com a equação 8, que implica na consulta a um sistema fuzzy para obter cada ângulo de mudança de orientação de cada *boi*d (como mostra a equação 7). A taxa de frames por segundo, que mantém-se estável em 60fps com a modelagem *crisp* clássica, cai para cerca de 20fps quando muitos agentes estão próximos (em testes com 50 agentes).

Portanto, trabalhos futuros devem avaliar formas de otimização, por exemplo: testar outras bibliotecas fuzzy, evitar o recálculo de valores, ou utilizar quadrees [9] para subdividir o espaço analisado.

2.5 Variações sobre os parâmetros

Já que a distância de outros *boi*ds não é utilizada para determinar a força de alinhamento, verificamos qual seria o resultado de manter apenas a posição relativa dos *flockmates* como variável de entrada para as regras de inferência de coesão e separação, e se ainda assim o comportamento simulado pelo sistema assemelharia-se ao real. Dessa forma, simplificamos as regras para:

Coesão:

1. IF position IS front THEN headingChange IS none;
2. IF position IS leftOrBehind THEN headingChange IS turnLeft;
3. IF position IS rightOrBehind THEN headingChange IS turnRight;

Separação:

1. IF position IS behind THEN headingChange IS none;
2. IF position IS leftOrFront THEN headingChange IS turnRight;
3. IF position IS rightOrFront THEN headingChange IS turnLeft;

Verificamos que o uso do método de Larsen (max-produto) não altera a saída para as regras utilizadas, e analisamos como outras formas de defuzzificação se comportam (veja a Figura 13 na seção Anexos). Ressalta-se que as funções de pertinência da variável de saída **headingChange** englobam o intervalo $[-360, 360]$ pois assim conseguimos uma saída defuzzificada no intervalo $[-180, 180]$ desejado, ao aplicar o método do centróide (Figura 14a) [10].

3 Linguagem de Programação

Utilizou-se a linguagem Kotlin para desenvolver o projeto, pela sua interoperabilidade com Java. Adotou-se a biblioteca jFuzzyLogic [11] para implementar o sistema fuzzy, e Processing [12] para criar a parte gráfica da simulação, pois ela foi desenvolvida de forma a apresentar uma maneira simples de programar no contexto de artes eletrônicas e visuais.

4 Resultados

Os dados foram coletados para duas configurações iniciais de posicionamento de 25 agentes: (I) todos no centro da tela (II) cada um com uma posição aleatória (conforme mostra a Figura 7). O tamanho da janela de simulação é de 720×720 pixels, e os parâmetros usados são apresentados na Tabela 1.

Parâmetro	ω_a	ω_c	ω_s	R^a	R^c	R^s	v_M	f_M
Unidade	-	-	-	px	px	px	px / frame	px / frame ²
Valor	0,3	1,2	1,4	80	80	30	2	1

Tabela 1: Parâmetros utilizados

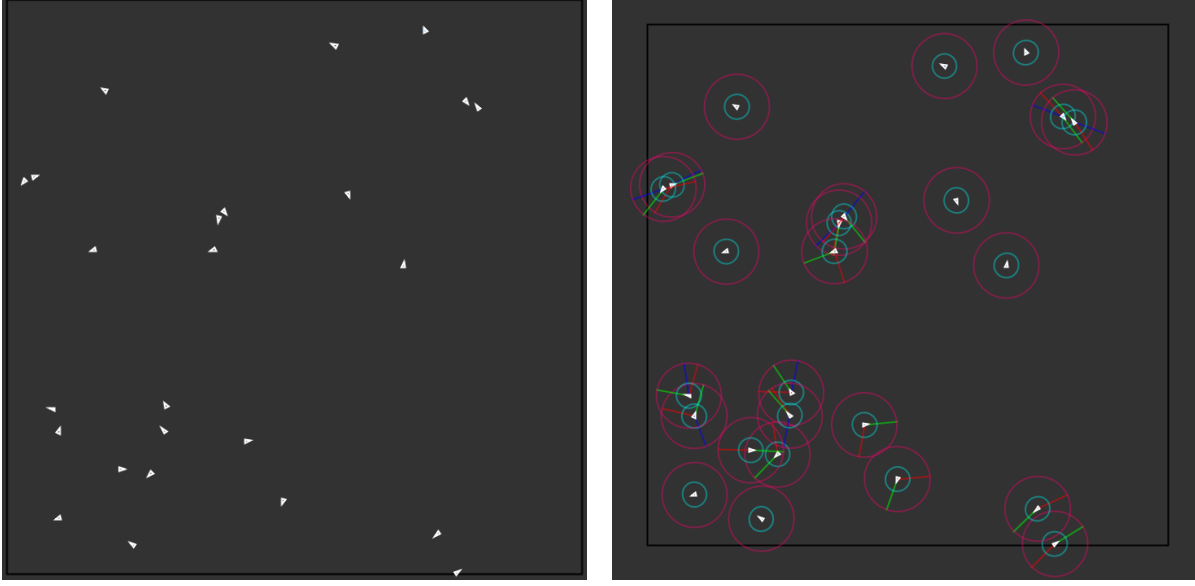


Figura 7: Configuração inicial com posições distribuídas aleatoriamente (esquerda) e visualização (direita) dos raios de separação (ciano), coesão e alinhamento (rosa), e das forças resultantes iniciais de alinhamento (vermelho), de coesão (verde) e de separação (azul)

Mediu-se as distâncias máxima e mínima entre *boids*. Para evitar saltos no valor da distância máxima quando um agente passa de um lado ao outro da tela, evitou-se *wraparound* multiplicando a velocidade de *boids* que estavam próximos de sair da janela de simulação por -1 de forma adequada para que eles ficassem confinados nela.

Os resultados são mostrados nas Figuras 8 e 9, onde a distância mínima média entre qualquer par de agentes é denotada por uma linha contínua azul e a máxima por uma linha verde. As regiões sombreadas coloridas mostram o intervalo de valores observado para as distâncias mínima e máxima a cada frame.

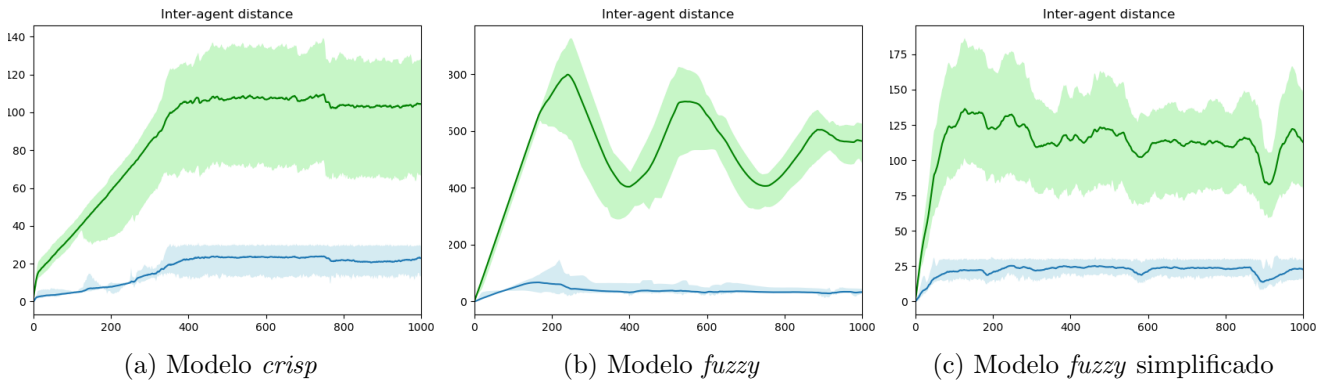


Figura 8: Distâncias entre agentes para a configuração (I)

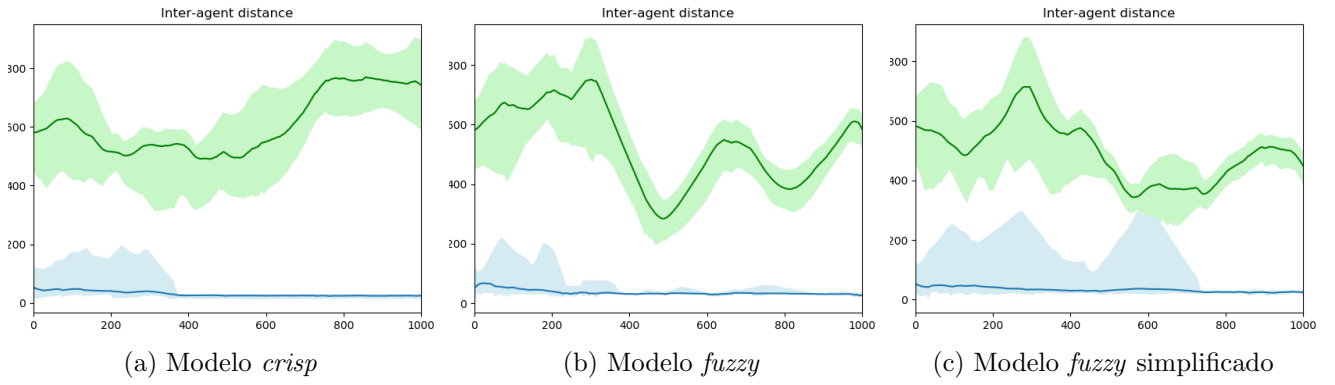


Figura 9: Distâncias entre agentes para a configuração (II)

Como o objetivo de uma simulação é imitar um processo real, uma boa implementação deve lembrar quem a assiste do comportamento real de *flocking*. Portanto, julgamos a avaliação subjetiva de pessoas não envolvidas no projeto como uma métrica que deve ser considerada ao determinar se o modelo fuzzy desenvolvido é equiparável ao modelo clássico.

Para isso, criamos um formulário onde mostramos um vídeo com 3 simulações² (veja a Figura 10), uma com o modelo *crisp* e duas com modelos fuzzy (o original, da Seção 2.3, e o simplificado, descrito na Seção 2.4).

Pede-se então que os participantes ordenem as implementações vistas pela sua semelhança com o comportamento real (tendo um vídeo³ como referência), e também que avaliem quão "inteligente" parece ser a forma como os *boids* se movem, utilizando a escala Likert [13].

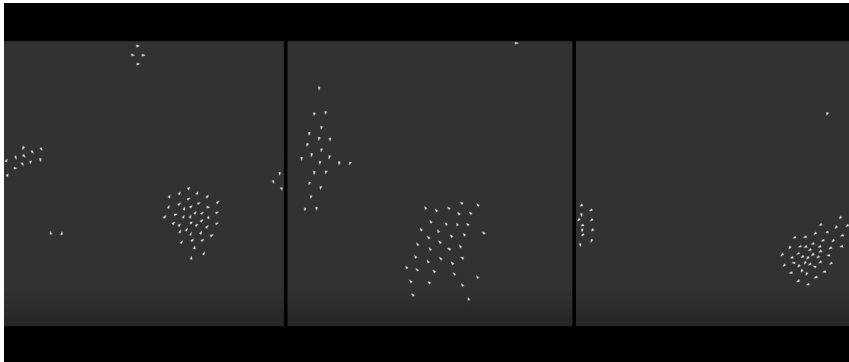


Figura 10: Frame do vídeo de 40 segundos apresentado no formulário de pesquisa, com os modelos: *fuzzy* simplificado (esquerda), *fuzzy* (centro) e *crisp* (direita)

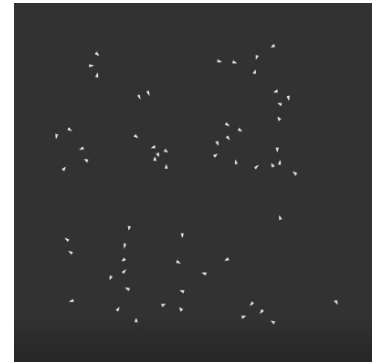


Figura 11: Configuração inicial, com 60 agentes e os mesmos parâmetros da Tabela 1

É importante ressaltar que os participantes não sabiam de que simulação era cada vídeo, além de não ter sido dada nenhuma explicação fora a que estava no formulário (Figura 14).

Doze pessoas responderam ao questionário, os resultados são apresentados abaixo, nas Figuras 12 e 13.

²<https://vimeo.com/341369793>, ou https://www.youtube.com/watch?v=svtQu_gInY8

³https://www.youtube.com/watch?v=V4f_1r80RY

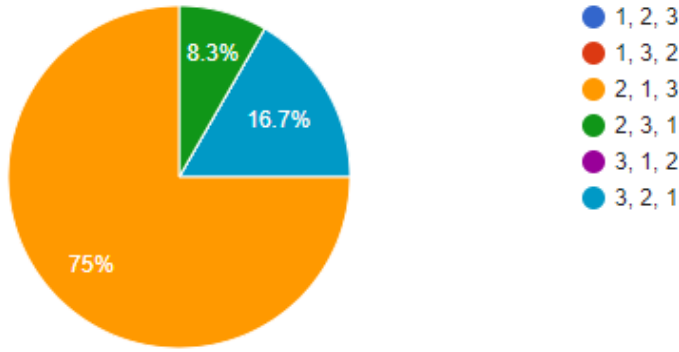


Figura 12: Resultados da questão: "Considerando as simulações apresentadas, ordene-as começando pela que mais lembra o comportamento real e terminando com a que menos se assemelha a ele."

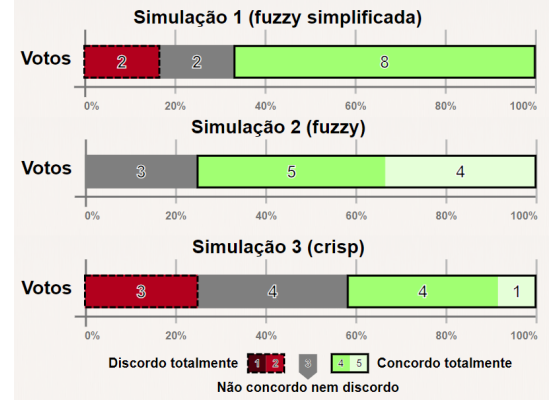


Figura 13: Resultados da questão: "Em relação a cada simulação, avalie a proposição: 'O comportamento apresentado parece inteligente/coordenado/complexo'."

5 Análise e discussão

Inicialmente, analisando os três gráficos obtidos, do modelo *crisp* clássico, do modelo *fuzzy* e do modelo *fuzzy* simplificado, podemos perceber que o modelo *fuzzy* simplificado e o modelo clássico são os que mais se assemelham, principalmente na configuração (II), onde ambos mantiveram um valor aproximadamente constante de distância máxima (em verde), porém de forma oscilante. Isso se dá ao fato de eles estarem constantemente fazendo correções para se manter na situação ideal de coesão, separação e alinhamento.

Já o gráfico do modelo *fuzzy* se comportou de maneira mais suave, sem fazer correções bruscas. Pode-se então atribuir o resultado mais natural encontrado com o modelo *fuzzy* a isso (vide Figura 12), pois ao invés de realizar alterações bruscas de direcionamento, como vemos no modelo clássico, a força de alinhamento definida pela equação 8 leva a uma mudança de forma mais gradual.

Na configuração (II), os indivíduos foram inicializados em posições aleatórias no canvas, já na configuração (I), todos foram inicializados no centro. Pode-se ver intuitivamente isso nos resultados dos máximos dos mínimos (área em azul claro a cima da linha de distâncias mínimas), onde na configuração (II) o maior valor de distância mínima entre agentes começou alto e então se estabilizou numa distância mais próxima da média dos mínimos, enquanto que na configuração (I) ele começa em 0, como a média dos mínimos, e se manteve relativamente constante durante todo o período de amostragem de 1000 frames.

Pode-se também observar que no caso do modelo *fuzzy*, o mínimo dos máximos e o máximo dos máximos se mantiveram mais próximos da linha de média das distâncias máximas (representados pela área em verde claro nos gráficos), em relação aos modelos *crisp* e *fuzzy* simplificado. Pode-se concluir a partir disso que o modelo *fuzzy* manteve uma distância mais uniforme dos indivíduos durante o período de amostragem, e isso é um comportamento desejável no problema de *flocking*.

Em ambas as configurações, observou-se que a média da distância mínima tendeu a estabilizar-se no valor 60 pixels para todos os gráficos, que é equivalente à duas vezes o raio de separação usado ($R^s = 30$), assim, todos os modelos são capazes de manter uma distância mínima de separação entre cada indivíduo.

Pelos respostas ao questionário (Figuras 12) vemos um resultado que superou nossas expectativas, pois 83% das pessoas (10 em 12) classificaram o modelo implementado como uma simulação mais semelhante ao comportamento real do que o modelo clássico baseado no algoritmo de Reynolds [5], sendo que tínhamos proposto como objetivo inicial apenas conseguir, com o uso de lógica

fuzzy, simular o comportamento de *flocking* de forma tão boa quanto com o modelo *crisp*.

Vemos na Figura 13 que a simulação *crisp* foi a que mais obteve votos discordando parcialmente (na escala Likert) da proposição que "o comportamento apresentado parece inteligente/coordenado/complexo". Apesar da questão abrir espaço a diferentes interpretações do que significa cada termo (além de que os participantes precisaram extrapolar sua concepção do comportamento real em 3D para 2D), a propomos pois consideramos que uma boa imitação de um fenômeno real deve transmitir as mesmas percepções que ele.

Assim, conjecturamos que a simulação *crisp* foi a pior avaliada devido principalmente ao comportamento de alinhamento, porque a orientação desejada é representada pela subtração dos vetores de direção dos agentes (equação 4), levando a mudança mais rápidas e bruscas do que ao representar a força de alinhamento como um vetor com a orientação média das orientações de *boids* vizinhos (equação 8). Deste modo, os agentes simulados geram um comportamento que não é natural ao parecem vibrar com alta frequência (o que pode ser observado como um "ruído" aparente nas Figuras 8a e 9a, enquanto que a curva das Figuras 8b e 9b é muito mais suave).

6 Conclusões

Algoritmos de *flocking* possuem muitas aplicações além da simulação de pássaros. Eles são aplicados em simulações de comportamento de diversos tipos de animais, podendo simular cardumes de peixes, bandos de morcegos, enxames de insetos e até mesmo multidões de pessoas. Isso é particularmente útil em jogos⁴, quando deseja-se simular uma grande quantidade de agentes não controlados pelo jogador que possuam comportamento de coesão e alinhamento, porém usos são encontrados de filmes⁵ a modelagens de fluxo de opinião em redes sociais [14].

Assim, quase toda situação que envolva o movimento coletivo de agentes pode ser representada tendo *steering behaviors* como base, demonstrando a relevância de explorá-lo.

Uma expansão natural da simulação implementada seria considerar três dimensões ao invés de apenas duas como proposto, levando a comparações mais diretas com o comportamento real. Outra modificação interessante, para uma imitação mais fiel possivelmente, seria considerar apenas os 7 *boids* mais próximos de um agente para sua vizinhança, já que [3] mostra ser esse o número considerado em *flocks* de pássaros esturnídeos.

Por fim, ressaltamos os resultados da avaliação subjetiva sobre a fidelidade da simulação (Figuras 12 e 13), considerando que ela tenta representar um comportamento tridimensional em apenas duas dimensões. Das 12 respostas, 10 classificaram as modelagens *fuzzy* como mais semelhantes do comportamento real do que ao usarmos o algoritmo *crisp* clássico de Reynolds, mostrando que o uso de lógica *fuzzy* é viável e deve ser pesquisado para a simulação de *flocking*.

Devemos lembrar, entretanto, que essa análise é válida para os parâmetros definidos na Tabela 1, utilizados tanto para as modelagens *fuzzy* como para a *crisp*, de modo que uma afinação diferente destes valores para que cada abordagem pode levar a resultados análogos entre elas (o que é algo também relevante de ser estudado).

O código fonte pode ser encontrado em github.com/laurelkeys/ai-intro (na pasta *boids*).

Referências

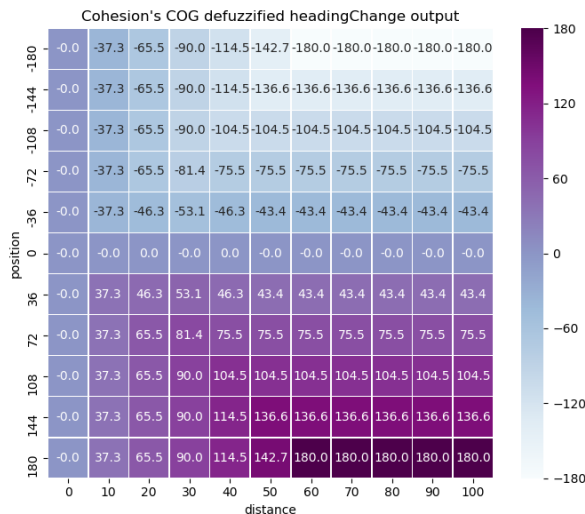
- [1] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, Aug. 1987. [Online]. Available: <http://doi.acm.org/10.1145/37402.37406>

⁴Flocking (Unity Wiki)

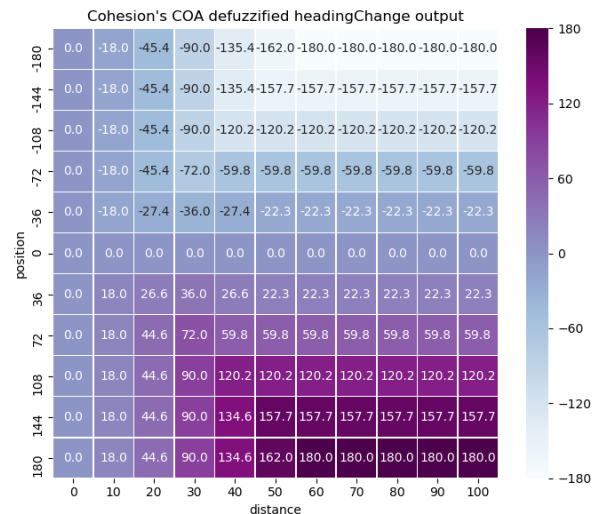
⁵Batman Returns (1992)

- [2] “Flocking (behavior) — Wikipedia, the free encyclopedia,” acessado em junho de 2019. [Online]. Available: [https://en.wikipedia.org/wiki/Flocking_\(behavior\)](https://en.wikipedia.org/wiki/Flocking_(behavior))
- [3] G. F. Young, L. Scardovi, A. Cavagna, I. Giardina, and N. E. Leonard, “Starling flock networks manage uncertainty in consensus at low cost,” *PLOS Computational Biology*, vol. 9, no. 1, pp. 1–7, 01 2013. [Online]. Available: <https://doi.org/10.1371/journal.pcbi.1002894>
- [4] C.-O. Erneholm, “Simulation of the flocking behavior of birds with the boids algorithm.” 2011, acessado em junho de 2019. [Online]. Available: http://www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2011/rapport/erneholm_carl-oscar_K11044.pdf
- [5] C. W. Reynolds, “Boids, background and update,” acessado em junho de 2019. [Online]. Available: <http://www.red3d.com/cwr/boids/>
- [6] E. L. Colombini, “Aula 3 – agentes,” acessado em junho de 2019. [Online]. Available: <https://www.ic.unicamp.br/~esther/teaching/2019s1/mc906/Aula4.pdf>
- [7] —, “Aula 4 – agentes do tipo problema e busca sem informação,” acessado em junho de 2019. [Online]. Available: <https://www.ic.unicamp.br/~esther/teaching/2019s1/mc906/Aula4.pdf>
- [8] B. Knapp, “Fuzzy sets and pattern recognition,” acessado em junho de 2019. [Online]. Available: <http://hci.sapp.org/lectures/knapp/fuzzy/fuzzy.pdf>
- [9] “Quadtree — Wikipedia, the free encyclopedia,” acessado em junho de 2019. [Online]. Available: <https://en.wikipedia.org/wiki/Quadtree>
- [10] I. Lebar Bajec, N. Zimic, and M. Mraz, “Simulating flocks on the wing: The fuzzy approach,” *Journal of theoretical biology*, vol. 233, pp. 199–220, 04 2005.
- [11] P. Cingolani and J. Alcala-Fdez, “jFuzzyLogic: A java library to design fuzzy logic controllers according to the standard for fuzzy control programming,” *International Journal of Computational Intelligence Systems*, vol. 6, pp. 61–75, 06 2013.
- [12] Processing Foundation, “Processing, version 3.3.6.” [Online]. Available: <https://processing.org>
- [13] “Likert scale — Wikipedia, the free encyclopedia,” acessado em junho de 2019. [Online]. Available: https://en.wikipedia.org/wiki/Likert_scale
- [14] S. Cole, “Modeling Opinion Flow in Humans Using Boids Algorithm & Social Network Analysis,” acessado em junho de 2019. [Online]. Available: https://www.gamasutra.com/view/feature/130232/modeling_opinion_flow_in_humans_.php

Anexos



(a) Centróide



(b) Bisseção

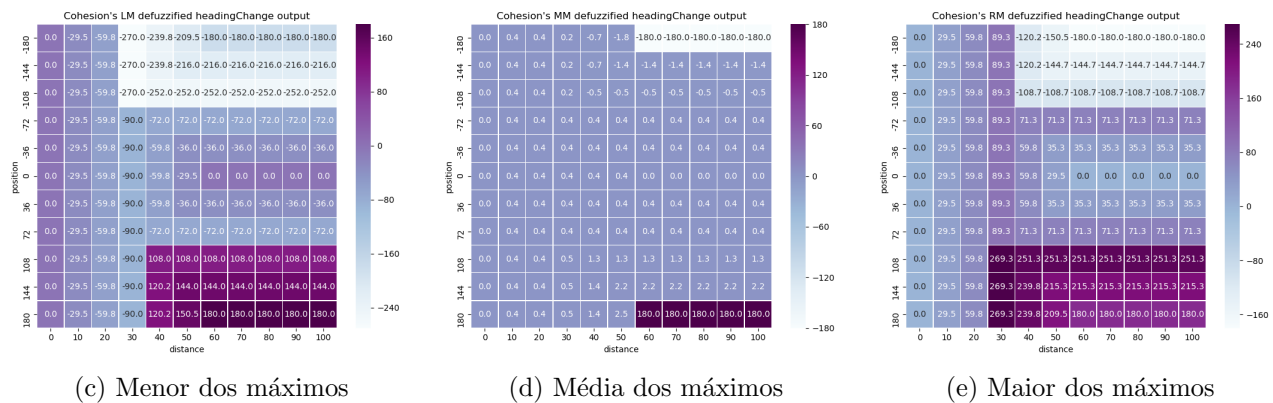


Figura 13: Saída para diferentes métodos de defuzificação

[MC906] Projeto 3

Neste projeto buscamos reimplementar uma simulação clássica do comportamento apresentado por grupos de pássaros ao voarem em conjunto, como mostra o primeiro vídeo abaixo (de forma similar a um cardume de peixes, ou a um enxame de abelhas). Assim, apresentamos no segundo vídeo 3 implementações distintas simuladas, partindo da mesma configuração inicial.

* Required

Comportamento real

Simulações [numeradas por: 1 (esquerda), 2 (centro) e 3 (direita)]

Considerando as simulações apresentadas, ordene-as começando pela que mais lembra o comportamento real e terminando com a que menos se assemelha a ele. *

☐ 1, 2, 3
☐ 1, 3, 2
☐ 2, 1, 3
☐ 2, 3, 1
☐ 3, 1, 2
☐ 3, 2, 1

Em relação a cada simulação, avalie a proposição: "O comportamento apresentado parece inteligente/coordenado/complexo". *

	1 (Discordo totalmente)	2 (Discordo parcialmente)	3 (Não concordo nem discordo)	4 (Concordo parcialmente)	5 (Concordo totalmente)
Simulação 1 (esquerda)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simulação 2 (centro)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simulação 3 (direita)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

SUBMIT

Figura 14: Formulário de pesquisa