

# Relatório de IAR

## Ant Clustering 1

Igor Schiessl Froehner<sup>1</sup>

<sup>1</sup>Centro de Ciências Tecnológicas, UDESC, Joinville.

`igor.sf14@edu.udesc.br`

### 1. Introdução

Dado a quantidade de dados geradas atualmente e o seu custo de armazenamento que vem baixando constantemente, o desafio atual é extrair-lhes valor. Para tal são propostas as mais diversas técnicas que visam obter insights, entender e prever comportamentos em várias instâncias e problemas que envolvem esses dados. Um dos problemas que é abordado pela área da inteligência artificial é o do "agrupamento". Conseguir extrair grupos de uma base de dados é de relevante utilidade para que se possa fazer descobertas sobre este [Huang 1998].

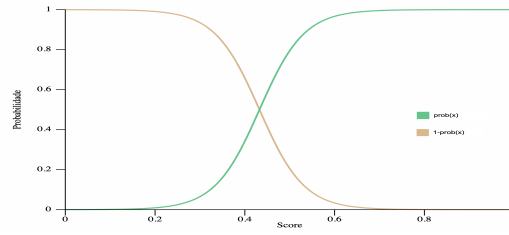
Dentro das técnicas que se fundamentam na evolução biológica, uma inteligência que vem sendo modelada computacionalmente é a de "enxames" de insetos [Jafar and Sivakumar 2010]. Por exemplo alguns comportamentos biológicos observados em formigueiros se aproximam muito do que consideramos agrupamento. Quando há formigas mortas no formigueiro, as formigas vivas tendem a agrupar-las para manter o formigueiro limpo. O modelo básico da modelagem desse comportamento se dá da seguinte forma: quando uma formiga encontra uma formiga morta ela tem uma chance de carregá-la; a partir do momento em que esta está carregando a formiga, ela tem uma chance de largá-la, mas a chance aumenta dependendo de quantas formigas mortas ela vê a sua volta [Jafar and Sivakumar 2010]. Dado uma quantidade de tempo, grupos de formigas mortas começarão a se formar.

Esse artigo busca estudar esse modelo básico do problema do agrupamento de formigas, estudando o modelo de probabilidade de pegar ou soltar as formigas e fazendo simulações visuais e numéricas. Este está organizado da seguinte forma: no capítulo 2 o desenvolvimento da simulação será explicado juntamente com a função de probabilidade utilizada; no capítulo 3 uma metodologia de experimentação será aplicada à simulação e os resultados serão analisados; e em 4 será feita uma breve conclusão sobre o trabalho desenvolvido.

### 2. Desenvolvimento

A simulação do ambiente e do problema foi feita computacionalmente usando a linguagem de programação Rust e uma Game Engine chamada bevy. O ambiente é um tabuleiro 2D quadrado de lado  $n$ , de  $n \times n$  células. As células das extremidades fazem fronteira também com as células do lado oposto, ou seja não há bordas. As formigas andam livremente e randomicamente pelo tabuleiro e elas tem um raio de visão usado para fazer o cálculo do *score*.

Dado o ambiente e os agentes, uma etapa importante do desenvolvimento foi determinar a função  $prob(x)$  a ser utilizada para determinar qual a probabilidade de uma formiga pegar ou largar uma formiga morta. Da seguinte forma:



**Figura 1. Gráfico da Função  $prob(x)$  e  $1 - prob(x)$**

1. Se a formiga não está carregada e não está sobre uma formiga morta ela se move para uma célula vizinha.
2. Se a formiga está carregada sobre uma célula vazia: ela calcula o *score* da célula em que se encontra.  $prob(score)$  será a chance dela soltar a formiga que carrega.
3. Se a formiga não está carregada sobre uma célula que tem uma formiga morta, ela calcula o *score* da célula em que se encontra e  $1 - prob(score)$  será a chance dela pegar a formiga.

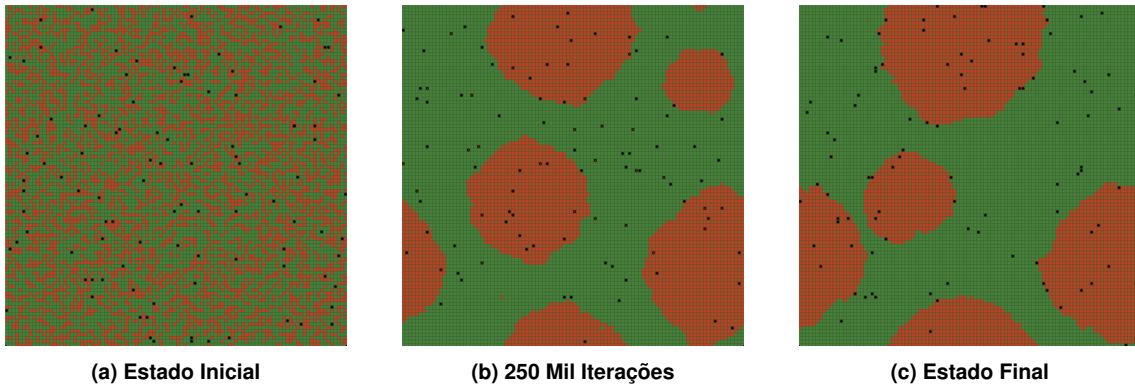
O *score* de uma célula é dado assim:  $score = \frac{nfm}{nc}$ , onde *nfm* é o número de células com formigas mortas que a formiga enxerga, e *nc* é o total de células que a formiga enxerga. Dado o *score* é possível calcular  $prob(score)$ , a função encontrada empiricamente que melhor teve desempenho é a seguinte (seu gráfico é dado na figura 1):

$$prob(x) = \frac{1}{1 + e^{-e^3(x - \frac{e}{2\pi})}} \quad (1)$$

Essa função foi encontrada pelo autor pois ele estava em busca de uma função que fosse parecida com a *sigmoid* [Monroe 2017] mas que fosse de 0 até 1 no intervalo com *x* indo também de 0 até 1. E também com uma crescente mais acentuada que a *sigmoid*, onde a sua elevação mais acentuada acontecesse perto do 0.5 para *x*.

### 3. Experimentos e Análise de Resultado

As variáveis que determinam a execução das simulações como o número máximo de iterações e o número de iterações por frame foram adaptados ao melhor valor para que um resultado conclusivo fosse atingido. A seguir encontra-se um exemplo de simulação com 5 milhões de iterações para as formigas com raio de visão 1:



(a) Estado Inicial

(b) 250 Mil Iterações

(c) Estado Final

**Figura 2. Exemplo de Simulação**

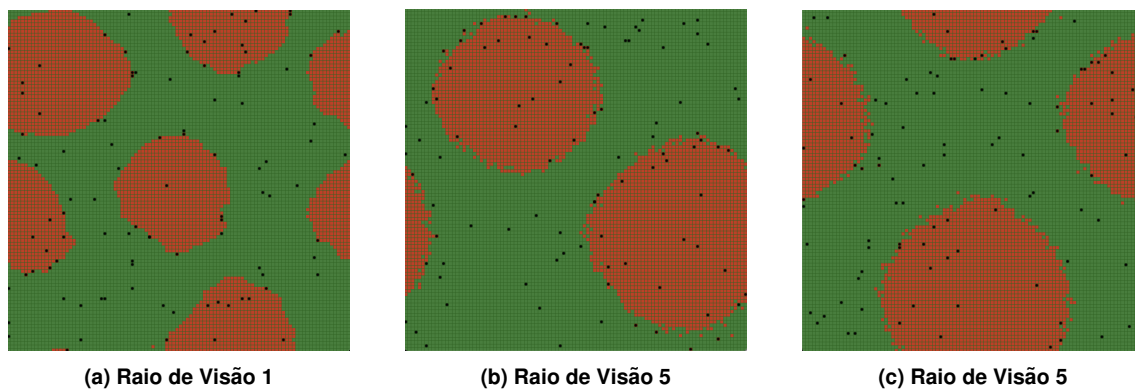
### 3.1. Metodologia

Para chegar a resultados mais conclusivos é importante colocar algumas métricas, no caso desse trabalho serão usadas duas métricas simples: maior agrupamento obtido e número de agrupamentos separados obtidos. Já que nesta etapa todos os itens são iguais, quanto maior for o maior agrupamento obtido e quanto menor for o número de agrupamentos melhor. Itens são considerados no mesmo agrupamento se uma faz fronteira com a outra, inclusive na diagonal.

O comparativos serão feitos quando ao raio de visão da formiga, para os valores 1 e 5. Serão feitos 5 testes para cada valor de raio de visão e estes testes farão 5 milhões de iterações. Demais parâmetros são: ambiente de 100 x 100 células, 4000 formigas mortas e 100 formigas vivas.

### 3.2. Análise dos Resultados

Nessa seção será feita a análise dos resultado através de uma tabela comparativa. Mas é interessante observarmos exemplos de execuções visuais da simulação, como a seguir onde (em 3a e 3b) é mostrado o estado final das simulações com 1 milhão de iterações:



**Figura 3. Exemplos de Execução para os Raios de Visão**

O resultado visto nessa imagem é melhor que o esperado para raio de visão 5, resultado provavelmente favorecido pela função escolhida. Mas é possível observar também que as bordas do agrupamento da simulação com raio 5 são mais esparsas, o que pode acabar sendo contabilizado com um agrupamento diferente. Por exemplo, em 3c, para uma execução com 1 milhão de iterações, visualmente temos somente um agrupamento, mas na métrica foram contabilizados 16.

	Maior Agrupamento	Número de Agrupamentos	Tempo de Execução
Raio 1	2256	4,2	16s
Raio 5	3262,8	15	141s

## 4. Conclusão

Tendo em vista os dados gerados através das métricas, o resultado foi diferente do esperado. Dado que as simulações com raio 5 tiveram um resultado melhor pela seguinte perspectiva: a contabilização do número de agrupamentos considerou somente as células vizinhas, e não a proximidade das células. Então visualmente a simulação tinha atingido

um único agrupamento, mas a métrica acabou contabilizando mais. Visualmente, no final de várias execuções foram criados somente um agrupamento para o raio 5. Já no quesito tempo, o raio influenciou muito no desempenho onde o aumento do tempo foi de quase 9 vezes.

## **Referências**

- Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304.
- Jafar, O. M. and Sivakumar, R. (2010). Ant-based clustering algorithms: A brief survey. *International journal of computer theory and engineering*, 2(5):787.
- Monroe, W. (2017). Logistic regression. *Recall*, 1(1).