

# Simulação Biológica de Agrupamento de Formigas Mortas

1<sup>st</sup> Marlon Henry Schweigert  
Departamento de Computação  
Centro de Ciências Tecnológicas - UDESC  
Joinville, Brasil  
marlon.henry@magrathealabs.com

2<sup>nd</sup> Rafael Stubs Parpinelli  
Departamento de Computação  
Centro de Ciências Tecnológicas - UDESC  
Joinville, Brasil  
rafael.parpinelli@udesc.br

**Resumo**—Este meta-artigo descreve o funcionamento computacional da simulação de formigas utilizando inteligência artificial dentro de um ambiente homogêneo visando simular o agrupamento de dados em lugares densos, realizando uma análise entre a variação dos parâmetros utilizados para o processamento da inteligência artificial.

**Index Terms**—formigas, inteligência artificial, modelagem matemática, programação concorrente

## I. INTRODUÇÃO

A busca por simular efeitos naturais que otimizem problemas em sistemas computacionais tem grandes vantagens comparados a algoritmos deterministas e objetivos, vistos que podemos dar sentido a resultados obtidos da natureza a qual algoritmos puramente matemáticos custam fazer qualquer sentido. Essa propriedade principal é aplicada ao algoritmo de aglomeração de formigas mortas por um grupo de formigas vivas.

O efeito de agrupamento de um ambiente poluído por formigas mortas foi descrito por Mohamed Jafar [1], a qual tem um sistema matemático pode descrever a movimentação de tais agentes [2].

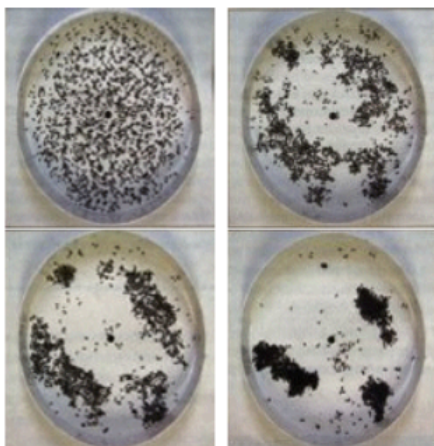


Figura 1. *Clusters* de formigas mortas.

Fonte: O. A. Mohamed Jafar, R. Sivakumar [1]

Tal comportamento é conhecido como agrupamento, na qual esse efeito pode ser utilizado para otimizar problemas de

agrupamento, limpeza ou desfragmentação de ambientes na qual os dados manipulados sejam homogêneos [1].

Este artigo será dividido em algumas seções:

- **Caracterização do problema:** Descreve a análise das características do problema e medidas de desempenho.
- **Modelagem matemática:** Descreve as ações e o modelo matemático a qual os agentes podem realizar.
- **Resultados obtidos:** Amostras estado de execução obtidos da modelagem descrita, assim como métricas de recursos utilizados.
- **Análise sobre resultados obtidos:** Análise sobre estados obtidos e modelagem descrita.

## II. CARACTERIZAÇÃO DO PROBLEMA

Para melhor entendimento do problema, faz necessário classificar o problema de agrupamento de formigas mortas. Para tal, precisamos definir quais suas características [1].

A sua classificação segue o determinado padrão:

- **Ambiente:** Matriz de formigas mortas e vivas.
- **Agente:** Formiga viva.
- **Sensores:** Analisar região em torno de si própria por um raio  $R$ .
- **Atuadores:** Pegar e soltar formigas mortas.
- **Desempenho:** Quantidade de grupos formados.

Algumas características dessa simulação são:

- Parcialmente observável;
- Estocástico;
- Sequencial;
- Dinâmico;
- Discreto;
- Multiagente;

## III. MODELAGEM MATEMÁTICA

O ambiente onde ocorrerá a simulação é desenhado como uma matriz esférica, ou seja, conectada pelas pontas. A fim de descrever matematicamente o comportamento de agrupamento de formigas mortas em tal ambiente, precisamos tomar alguns conceitos de como o fenômeno ocorre [1]:

- **Sem comunicação entre agentes:** As formigas não utilizam nenhum método de comunicação entre outras formigas vivas para realizar este comportamento.

- Esse comportamento sempre é executado visando melhorar o caminho do ambiente.
- As ações tomadas podem ser descritas como:
  - Pegar formiga;
  - Soltar formiga;
  - Caminhar;
- As únicas percepções que a formiga utiliza para tomada de decisão são:
  - Densidade da região onde ela está, utilizando um raio de visão.
  - Se em sua atual posição existe ou não uma formiga.

Dada essas características, podemos modelar as ações da formiga como uma cadeia de Markov a qual suas ações futuras requerem do seu estado atual. Essa cadeia de Markov é exibida na figura 2.

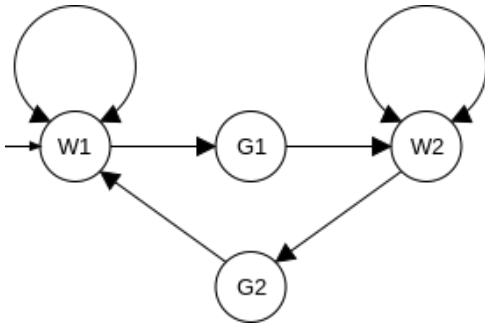


Figura 2. Autômato de estado de cada formiga.

Os estados descritos na figura 2 são:

- W1: Andando sem carregar uma formiga morta.
- G1: Pegar alguma formiga. Este estado tem uma probabilidade calculada dinamicamente conforme a posição atual da formiga.
- W2: Andando com uma formiga morta.
- G2: Soltar uma formiga morta. Este estado tem uma probabilidade calculada dinamicamente conforme o local da formiga.

Para este modelo funcionar, precisamos de um raio de visão ( $R$ ), a qual significa quantas casas ao seu redor uma formiga pode observar. Esse parâmetro pode ser observado na figura 3.

O fator de decisão principal para pegar e soltar uma formiga morta está em torno da variável  $D$  a qual é definida pela densidade local de cada formiga em torno do raio  $R$  de cada formiga [2].

$$D = \sum_{i=-R}^R \sum_{j=-R}^R A(i, j), \text{ onde}$$

$$A(i, j) = \begin{cases} 1, & \text{if } (has\_ant\_at(i, j)) \\ 0 & \end{cases}$$

A abordagem desta simulação utiliza uma memória para cada formiga onde é armazenado qual a maior densidade ( $D_{max}$ ) a qual esta formiga encontrou em sua vivência com o ambiente de testes. Essa informação é útil para normalizar dentro de um valor real  $[0, 1]$  a variável de Densidade relativa ( $D_{rel}$ ). A fórmula de normalização é dada por [2]:

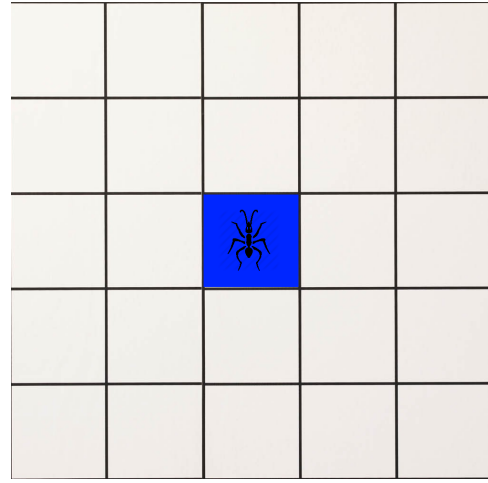


Figura 3. Formiga sobre a matriz, utilizando raio de visão 2.

$$D_{rel} = \frac{D}{D_{max}}$$

Por fim, o modo de pensar da formiga é dado por tal algoritmo:

```

def think():
    walk()
    if garbaging():
        if random() > relative_density():
            if cel_is_clear():
                put_dead_ant()
        else:
            if random() < relative_density():
                if not cel_is_clear():
                    get_dead_ant()

```

Tal simulação foi criada em Golang para ter alto desempenho, facilidade de escritas de imagens e programação paralela facilitada.

Para funcionamento geral, executamos algumas threads onde cada thread manipula um conjunto de formigas. Cada formiga executará o algoritmo descrito acima por  $n$  vezes. Ao final, caso alguma formiga esteja carregando alguma formiga morta, elas executam paços até encontrarem um local bom para soltar e deixam de executar.

#### IV. RESULTADOS OBTIDOS

Utilizando  $R = 1$  foram obtidos as amostras (IV), a qual demonstram o agrupamento funcional mesmo pouca informação global para cada agente. Em comparativo a amostra obtida com  $R = 5$  (IV), as bordas são bem mais definidas no teste executado com  $R = 1$ , porém em contra partida os grupos são mais definidos com  $R = 5$ .

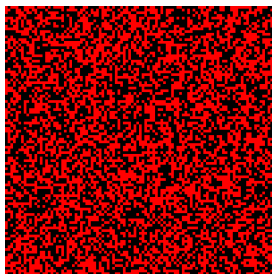


Figura 4. Estado inicial com  $R=1$

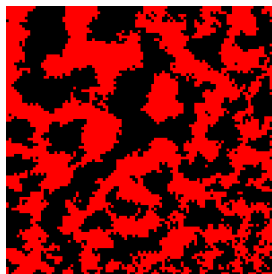


Figura 5. 100000 passos com  $R=1$

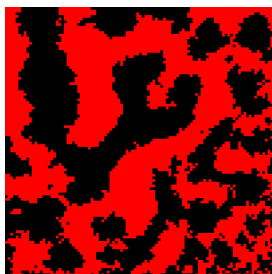


Figura 6. 300000 passos com  $R=1$

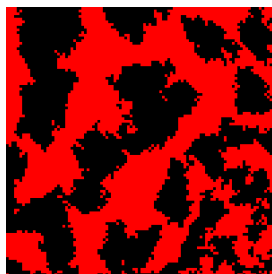


Figura 7. 500000 passos com  $R=1$

A regra de aglomeração utilizando  $R = 1$  torna as bordas bem delimitadas e propicia a criação de grupos menores. O resultado final cria caminhos pequenos e tortuosos utilizando  $R = 1$  (figura IV).



Figura 8. Estado inicial com  $R=5$

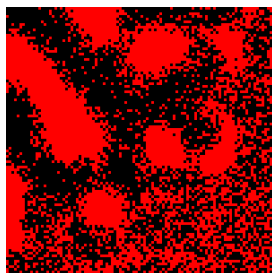


Figura 9. 100000 passos com  $R=5$

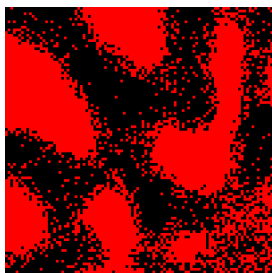


Figura 10. 300000 passos com  $R=5$

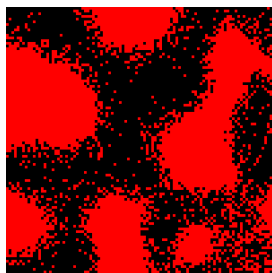


Figura 11. 500000 passos com  $R=5$

A regra de aglomeração utilizando  $R = 5$  (figura IV) proporciona uma maior fragmentação dos grupos, mas ainda

geram grupos bem definidos. Podemos perceber que a quantidade de passos para as formigas formar grupos definidos é maior comparado ao teste de  $R = 1$  (figura IV).

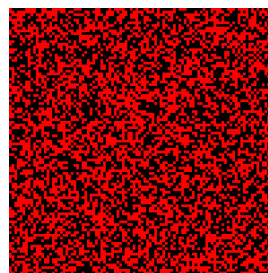


Figura 12. Estado inicial com  $R=10$

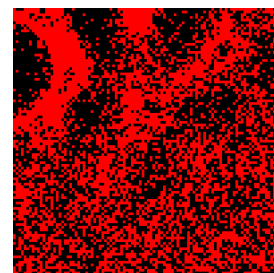


Figura 13. 100000 passos com  $R=10$

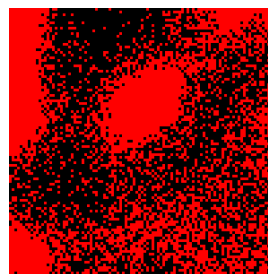


Figura 14. 300000 passos com  $R=10$

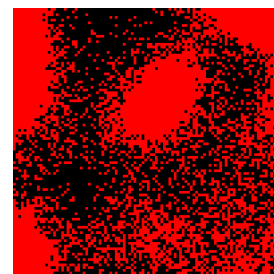


Figura 15. 500000 passos com  $R=10$

Podemos ver um agravante na fragmentação das bordas dos grupos utilizando  $R = 10$  (IV). Além disso, podemos perceber que o agrupamento nos primeiros passos, entre 0 e 100000 passos é muito mais custoso a ocorrer comparado aos raios menores.

Tabela I  
TABELA DE TEMPO DE EXECUÇÃO

	$R = 1$	$R = 5$	$R = 10$
Tempo médio de 10 execuções	4.442s	44.869s	2m 4.401s
variação	0.5%	0.9%	1.1%

Podemos assumir então, que quanto maior o raio, mais disperso será as bordas dos grupos. Isso é dado pela quebra de sistemas emergentes, visto que cada agente está levando em conta uma região muito grande ao invés de se preocupar com poucas informações. Essa comparação final é explicitada na figura IV.

Além do resultado final não ser satisfatório, o tempo de execução não é bom quanto o esperado, comparando o resultado de raios menores com raios maiores. A complexidade  $O(r)$  para verificação da existência de formiga morta em uma determinada casa da matriz dificulta o tempo de execução quanto maior seja o raio das formigas vivas.

## V. CONCLUSÃO

Podemos concluir que, em sistemas emergentes, passar mais informação a ponto de quebrar a ideia de sistemas emergentes acaba dificultando a convergência do sistema.

Comparativo entre estados finais com raios diferentes:

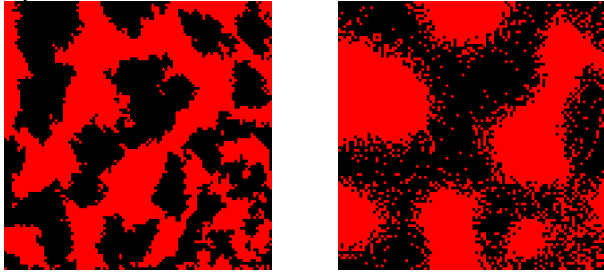


Figura 16. 500.000 passos com  $R = 1$  Figura 17. 500.000 passos com  $R = 5$

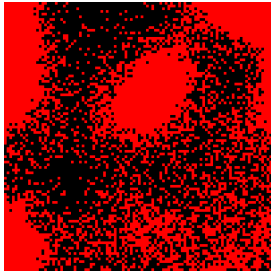


Figura 18. 500.000 passos com  $R = 10$

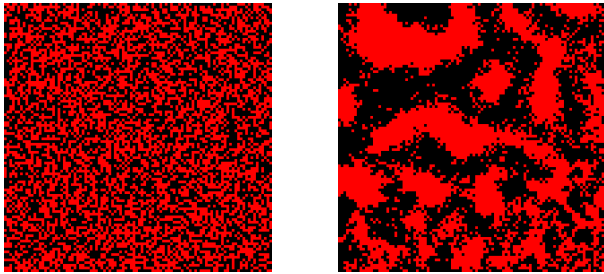


Figura 19. Estado inicial com  $R = rand(5) + 1$  Figura 20. 100000 passos com  $R = rand(5) + 1$

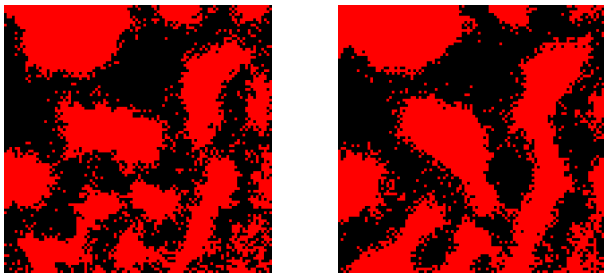


Figura 21. 300000 passos com  $R = rand(5) + 1$  Figura 22. 500000 passos com  $R = rand(5) + 1$

Percebe-se que a variação  $R \ni [1, 5]$  é um bom valor. O resultado obtido, com o raio variando dentro deste limite, é um bom resultado, conforme visto na figura V. Percebe-se que com essas configurações é criado grupos maiores, comparado

aos grupos criados com  $R = 5$ . A grande diferença é a fragmentação dos grupos nas bordas, que diminui por conta das formigas de menor raio.

A pesar dessa abordagem ser boa para formar grupos maiores e com uma fragmentação menor, faz necessário executar mais passos para remover a grande maioria das formigas espalhadas em torno dos grupos.

O tempo de execução utilizando tal abordagem de formigas homogêneas, utilizando 1.000.000 passos executa em 41.549s, com variação de 1.7%, sendo uma boa forma de melhorar o comportamento com essa modelagem matemática.

## REFERÊNCIAS

- [1] O. A. Mohamed Jafar, R. Sivakumar, "Ant-based Clustering Algorithms: A Brief Survey" Phil. International Journal of Computer Theory and Engineering, vol. 2, No. 5, October, 2010.
- [2] J. Handl, J. Knowles, M. Dorigo, "Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and 1d-som", Citeseer, 2003, pp.204–213.