

Análise da eficiência de diferentes técnicas de abordagem para a solução do problema de Agentes Ant Clustering

Guilherme Müller Utiana

Universidade do Estado de Santa Catarina (UDESC)

Joinville

`guilherme.utiana@edu.udesc.br`

Gustavo Daniel

Universidade do Estado de Santa Catarina (UDESC)

Joinville

`gustavo.daniel@edu.udesc.br`

RESUMO

No contexto cotidiano, é frequente lidar com problemas que a abundância de decisões e técnicas para otimizar o problema. Nesse trabalho, o problema de Ant clustering tem como objetivo amontoar uma quantidade de formigas mortas em um campo pré definido. O desenvolvimento do algoritmo teve resultados impressionantes, pois, se esperava algo pior por lidarmos com um problema e solução estocástica.

PALAVRAS CHAVE. Ant clustering. Otimização. Estocástica.

1. Introdução

Esse trabalho tem como objetivo o desenvolvimento de um algoritmo para resolver o problema *ant clustering*. No algoritmo não foi utilizado nenhum algoritmo como base para o desenvolvimento, isto é, todas as ideias de caminhos, decisões e critério de parada, foram ideias dos autores.

Em resumo, o problema tem como objetivo, agrupar itens que estão espalhados num campo qualquer, este campo é definido computacionalmente por uma matriz bidimensional. Esses itens são formigas falecidas, e quem carregará essas formigas mortas são formigas vivas, havendo nesse contexto mais formigas mortas do que vivas. A quantidade de montes formados por itens não é definido, assim, poderá ter desde 1 monte de grupos, 2, 3 e assim por diante.

2. Metodologia de Desenvolvimento

O trabalho consiste em 3 etapas, a primeira é o estudo da linguagem utilizada para o desenvolvimento, neste caso foi utilizado a linguagem Python 3.8. A segunda etapa consistiu na decisão de atributos e parâmetros para as classes desenvolvidas. E por fim, foi o próprio desenvolvimento e experimentos, até chegar à um refinamento de parâmetros.

A primeira etapa, consistiu-se de relembrar alguns conceitos de python, não se tiveram muitos problemas em relação à linguagem, pois, os autores já conheciam a linguagem, e já tinham trabalhado com a mesma.

A segunda etapa consistiu em criar duas classes, uma sendo o nosso campo, ou seja, a matriz que continha os agentes, os itens, os operadores, o tamanho do campo. Já na classe dos agentes, possuem-se os atributos da localização do agente no mapa, ou seja, a coordenada x e y, também existe atributo de visão, isto é, mostra o raio de visão que um agente possui a partir da localização dele, e também existe o atributo de carregamento, este atributo representa se um agente está carregando ou não um item.

Os métodos usados na classe dos Agentes para a interação com o ambiente consistem em:

- mover: deslocamento aleatório do agente para até uma casa de distância.
- verificar arredores: verifica quantos itens há nos arredores e a proporção deles para o perímetro analisado.
- interagir: com base nos dados obtidos ao verificar os arredores, decide qual o próximo passo do agente, se é pegar um item, soltar um item ou só se mover.

3. Descrição de Experimentos/Simulações e Resultados Obtidos

Neste trabalho foram feitos experimentos visando simular o comportamento das formigas ao se deparar com uma situação onde seu ambiente está com muitos obstáculos, ou seja, caótico onde estas então começam a agrupar os obstáculos em um ponto com a finalidade de liberar espaço para uma melhor movimentação.

No caso, foi feita uma simulação de tal comportamento utilizando uma matriz para representar o ambiente onde estão as formigas (agentes a partir de agora) e os obstáculos (itens a partir de agora), sendo colocados na matriz de forma aleatória, a partir deste ponto, foram feitas iterações entre todos os agentes para interagirem com o ambiente e dessa forma agruparem os itens para liberar espaços sem itens no ambiente.

Em geral, foram realizados dois métodos de experimentação, uma visando o algoritmo desenvolvido pelos autores sem base de referência sobre quaisquer algoritmos que resolvessem o problema. A outra etapa foram realizadas experimentos com base nos algoritmos propostos pelo professor, utilizando técnicas de cálculo para verificar a vizinha com o método da distância Euclidiana, e a outra técnica é a dissimilaridade entre pares de dados recebidos por um arquivo, utilizando o cálculo da distância Euclidiana.

A interação ocorria da seguinte forma, o agente escaneia seus arredores, tendo um campo de visão de 1, 3 ou 5 casas de distância do ponto onde se encontra, isto é, na matriz, vai formar um quadrado 3x3, 7x7 ou então 11x11, ao escanear seus arredores, consegue saber quantos itens tem nessa região, com base nisso e no fato de estar ou não deslocando algum item, faz uma das seguintes escolhas:

- pegar um item, caso tenha poucos itens nos arredores e não esteja deslocando algum item e tenha algum item para pegar;
- soltar um item, caso tenha muitos itens nos arredores e esteja deslocando algum item e no lugar atual não tenha nenhum item; e
- continuar a se mover caso nenhum dos dois acima.

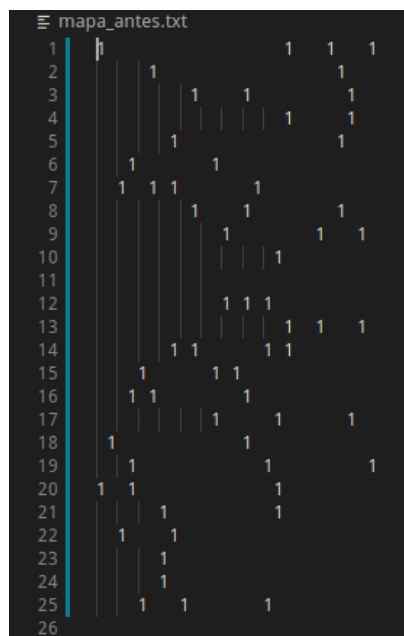
Os resultados obtidos foram salvos em arquivos contendo uma descrição da matriz com os locais onde se encontravam todos os itens inicialmente e após a execução do algoritmo.

Para a segunda versão do código em que os itens não são mais homogêneos (sendo agora cada um dos itens um conjunto de valores), foram feitas alterações no modo como os agentes lidam com tais itens, sendo agora necessário o uso de fórmulas mais específicas para a probabilidade de pegar e soltar os mesmos, tais fórmulas foram propostas pelo professor. Todos os testes realizados teve como condição de parada a quantidade de iterações.

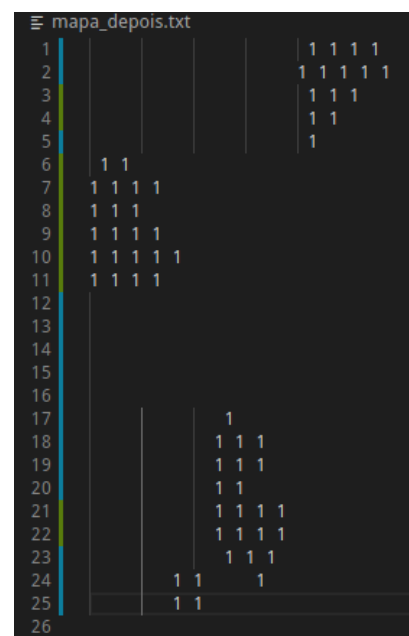
4. Análise dos Resultados Obtidos

Resultados obtidos foram satisfatórios, tendo testado com matrizes de diferentes tamanhos (25x25 e 50x50) e diferentes porcentagens de ocupação da matriz por itens e por agentes. É notável a diferença entre a matriz inicialmente e após a execução do algoritmo. As imagens 1 e 2 foram os resultados obtidos do algoritmo inicialmente proposto, sem base de algoritmos que já resolvem o problema do Ant Clustering.

Nas sub figuras 1a e 1b representam respectivamente o mapa inicial e o mapa após a execução do algoritmo. Nesse experimento, foram utilizados os parâmetro de 20% de preenchimento de itens no mapa, 5% de agentes, tamanho da visão dos agentes foi de 1, o tamanho do mapa sendo 25x25 e rodando com 100.000 iterações.



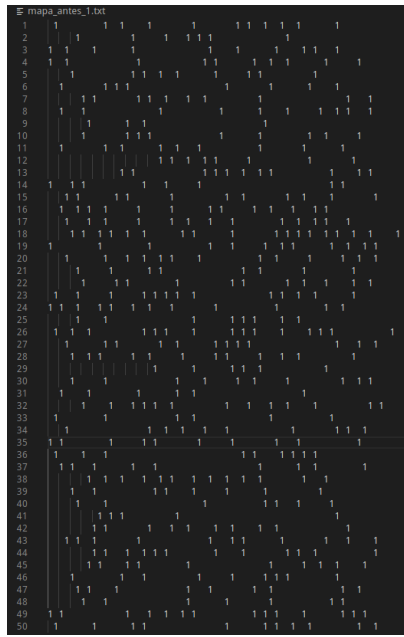
(a) Mapa gerado antes da aplicação do algoritmo



(b) Mapa gerado após a aplicação do algoritmo

Figura 1: Desempenho do algoritmo com mapa do tamanho 25x25 e visão 1

Nas sub figuras 2a e 2b representam os resultados obtidos. Nesse experimento, foram utilizados os parâmetro de 50% de preenchimento de itens no mapa, 7% de agentes espalhados no mapa, o tamanho da visão dos agentes foi definido com o valor de raio 3, o tamanho do mapa sendo 50x50 e tendo 100.000 iterações.



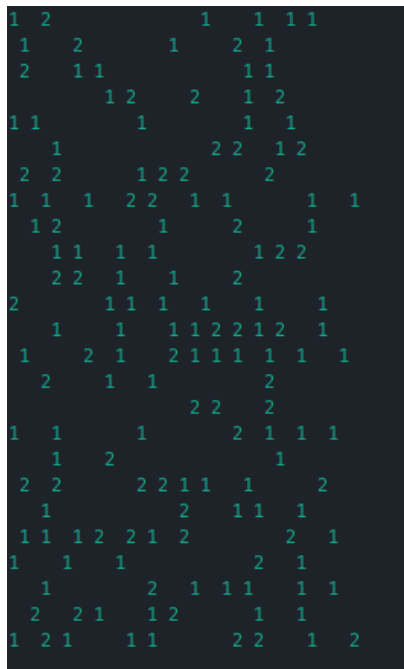
(a) Mapa gerado antes da aplicação do algoritmo



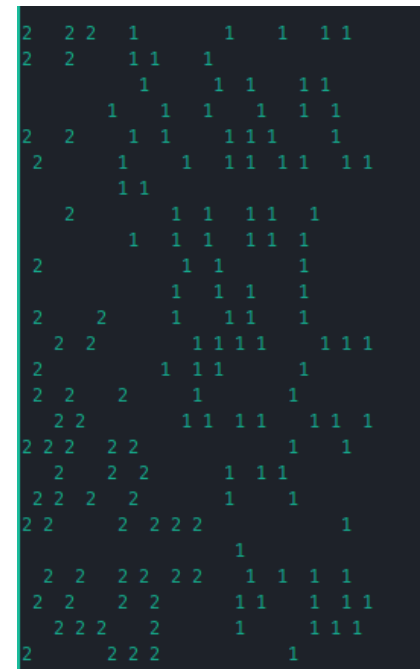
(b) Mapa gerado após a aplicação do algoritmo

Figura 2: Desempenho do algoritmo com mapa de tamanho 50x50 e visão 1

Nas próximas figuras, foram utilizados o método proposto pelo professor. Nas figuras 3 e 4 foi definido o tamanho da matriz sendo 25x25. A quantidade de iterações foi de 100.000, com taxa de 40% de preenchimento de itens, uma taxa de 5% de agentes espalhados pelo mapa, e o raio de visão de tamanho 1.



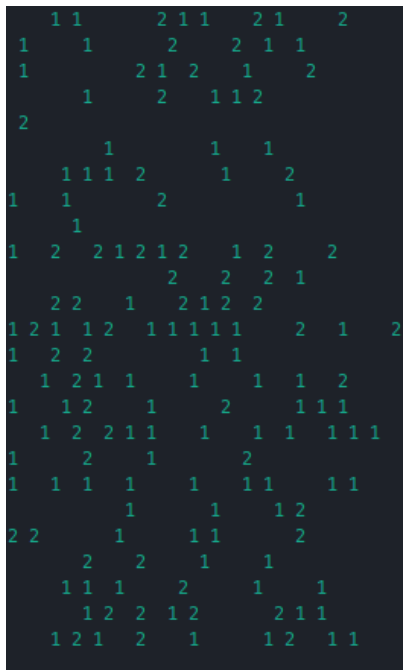
(a) Mapa gerado antes da aplicação do algoritmo



(b) Mapa gerado após a aplicação do algoritmo

Figura 3: Desempenho do algoritmo com mapa de tamanho 25x25 e visão 1

Já nas sub figuras 7a e 7b foi definido o tamanho da matriz de tamanho 50x50, uma



(a) Mapa gerado antes da aplicação do algoritmo



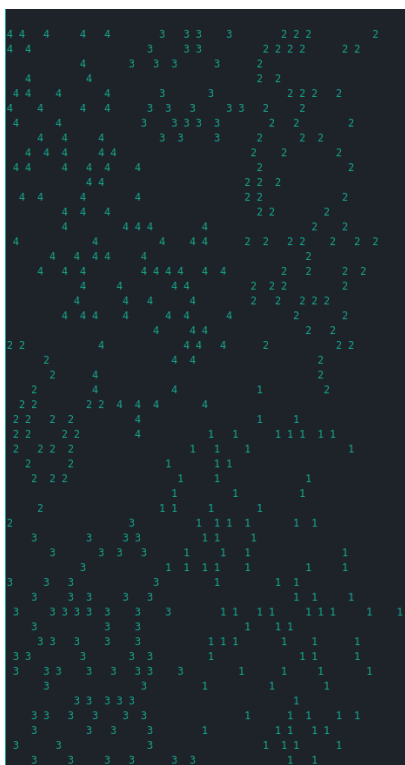
(b) Mapa gerado após a aplicação do algoritmo

Figura 4: Desempenho do algoritmo com mapa de tamanho 25x25 e visão 3

quantidade de 100.000 iterações, preenchimento de 40% de itens na matriz, 5% de agentes, e o raio de visão sendo 5.



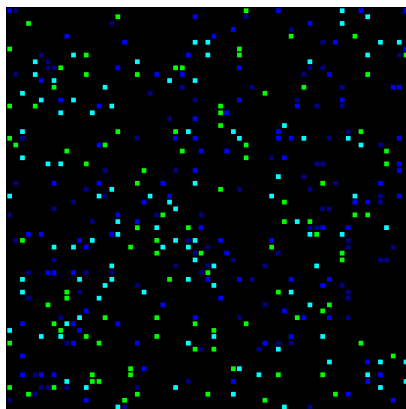
(a) Mapa gerado antes da aplicação do algoritmo



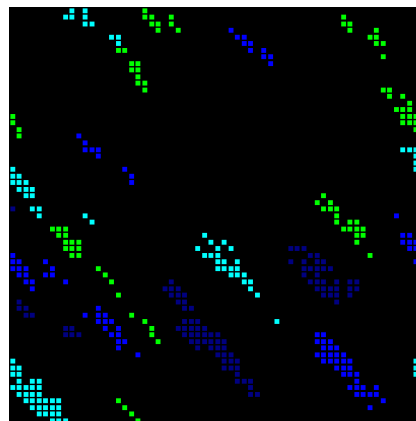
(b) Mapa gerado após a aplicação do algoritmo

Figura 5: Desempenho do algoritmo com mapa de tamanho 50x50 e visão 5

Por fim, foram realizados os testes do mesmo algoritmo proposto pelo professor, porém, foram adicionados interface gráfica em que, cada pixel representa um tipo de dado. A figura 6 representa os dados com 4 tipos de itens, e a figura 7 representa o tipo de dado com 15. Lembrando que, alguns parâmetros foram mantidos, os alterados foram: 4.000 iterações do algoritmo, 10 agentes fixos para os mapas, e raio de visão 1.

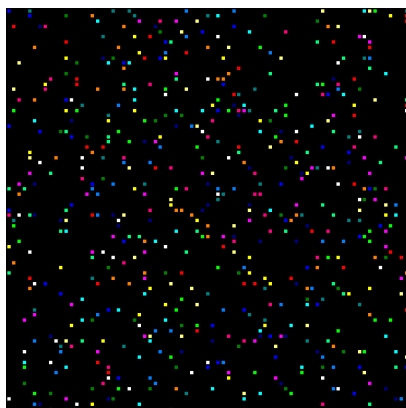


(a) Mapa gerado antes da aplicação do algoritmo

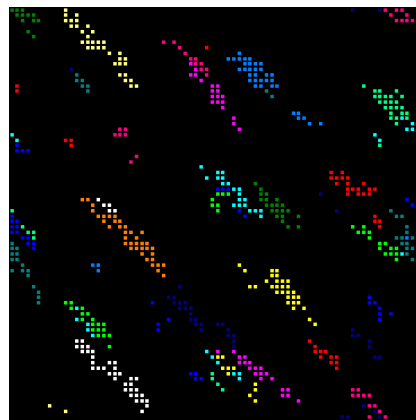


(b) Mapa gerado após a aplicação do algoritmo

Figura 6: Desempenho do algoritmo com 4 grupos e visão 1



(a) Mapa gerado antes da aplicação do algoritmo



(b) Mapa gerado após a aplicação do algoritmo

Figura 7: Desempenho do algoritmo com 15 grupos e visão 1

5. Conclusões e Trabalhos Futuros

Este trabalho teve como objetivo conceituar a modalidade de algoritmos estocásticos, consequentemente, viu-se que na seção 3 o resultado foi excelente, porém, na próxima execução, o resultado pode não ser tão bom quanto. Assim, para otimizar o desempenho e talvez melhorar os resultados, utiliza-se técnicas não exatas, isto é, heurísticas. Por ser um algoritmo com base de tomadas de decisões totalmente randômicas, poderá apresentar quaisquer resultados, não satisfazendo alguns usuários, isto ocorre, pela maioria dos casos de desenvolvimento de algoritmos, utilizarmos técnicas exatas, trazendo resultados ótimos.

Contudo, as técnicas utilizadas pelo professor, foram de grandes utilidades, resolvendo entradas mais genéricas, envolvendo dados, e separando itens por tipos, isto é, agrupando-os pelas suas categorias. Como trabalhos futuros, seria de grande aprendizado mostrar a comparação em

relação à otimização do desempenho e resultados obtidos, na aplicação desse algoritmo com uma outra heurística. Também pode-se melhorar este algoritmo com estratégias de decisões diferentes.

Referências

- Jafar, O. M. e Sivakumar, R. (2010). Ant-based clustering algorithms: A brief survey. *International journal of computer theory and engineering*, 2(5):787.
- Villwock, R., Steiner, M. T. A., e Siqueira, P. H. (2012). Pattern clustering via ants colony, ward method and kohonen maps. *International Journal of Computer Science and Network Security (IJCSNS)*, 12(6):81.