

Relatório o primeiro trabalho de Sistemas Inteligentes

João Pedro Makuch de Souza¹, Marcos Eduardo Travasso¹

¹Departamento Acadêmico de Informática
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba – PR – Brasil

{joao.250800,mtravasso}@alunos.utfpr.edu.br

1. Metodologia

1.1. Ambiente

Para o problema do cenário de pós acidente, temos um ambiente com o estado sendo as posições do mapa, onde o agente pode estar em uma, assim como vítimas e paredes, de forma com que o agente não possa estar na mesma posição de uma parede, mas possa estar na posição de uma vítima, onde a única mudança do estado do ambiente é realizada pelo deslocamento do agente, sendo assim um ambiente estático e determinístico.

O ambiente então é parcialmente observável, pois os agentes vão construindo com base nas suas percepções das posições visitadas o mapa visível para eles, e no final esses mapas são unificados em um só mapa que pode ou não conter todas as posições disponíveis.

Multiagente, pois além de dois grupos de agentes, cada grupo contém quatro agentes que trabalham de maneira cooperativa para encontrar e resgatar o maior número de vítimas no menor tempo possível, tornando também semidinâmico, pois o ambiente não irá mudar, mas a medida de desempenho de salvar o maior número de vítimas muda.

Os agentes também possuem uma bateria individual, de maneira com que as suas ações sejam sequenciais para o melhor aproveitamento da bateria, e discreto pois há um número finito de percepções e ações possíveis para os agentes.

1.2. Estratégia

A estratégia inserida nos agentes foi de uma busca circular ao redor da posição atual, funcionando como uma espécie de busca em profundidade, priorizando sempre as posições que não foram visitadas antes, mas sem deixar de registrar na sua memória as outras posições possíveis, dessa forma, quando o agente se encontra preso em posições já visitadas como na Figura 1 ele é capaz de buscar na sua memória alguma posição não visitada e se encaminhar para ela. Caso não exista uma posição livre, ou seja, ele visitou todas as posições alcançáveis, ele seguirá o caminho de retorno até a base.

Os quatro agentes exploradores receberam quatro estratégias de busca para a próxima posição não visitada, consistindo em inícios diferentes da mesma ordem de verificação:

NORTE → NORDESTE → LESTE → SUDESTE → SUL → SUDOESTE → OESTE → NOROESTE

Dessa maneira, cada agente irá iniciar a sua verificação de próxima posição iniciando por um dos pontos cardeais (norte, sul, leste, oeste), e seguindo em sentido horário, sempre escolhendo como próxima posição a primeira posição não visitada que encontrar.

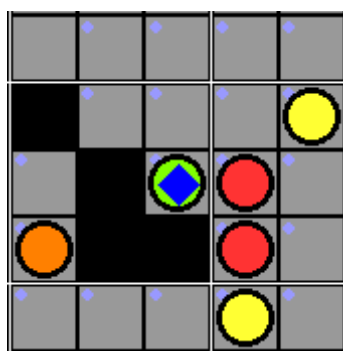


Figura 1. Agente preso entre posições já visitadas

O algoritmo do mapa dos agentes utilizado para criar os caminhos entre dois pontos é o A^* , para que dessa forma o agente possa maximizar o seu tempo restante considerando sempre o menor caminho possível conhecido entre a sua posição e o seu destino.

Essa estratégia de busca foi adotada, pois assim garantimos que os agentes irão para direções diferentes, aumentando a área vasculhada. Também tentamos uma estratégia similar em que os agentes seguissem com prioridade para um ponto cardeal, priorizando os pontos colaterais relativos a eles, mas essa abordagem não apresentou uma vantagem significativa se comparada com a busca sequencial no sentido horário dos pontos cardiais nos ambientes testados.

1.3. Deliberação

Em todo momento de deliberação, o agente irá verificar quatro casos:

- Se a posição atual corresponde a base e ele estava no caminho de retorno: nesse caso o agente irá alertar o socorrista "chefe" que será responsável por unir os mapas e dividir as vítimas encontradas.
- Se está no caminho de retorno: nesse caso ele continuará seguindo o seu caminho para voltar para a base.
- Se o tempo necessário para voltar para base desde a última verificação será menor do que o tempo restante de bateria: nesse caso o agente irá calcular o custo relativo a posição atual até a base, levando a mais dois casos:
 - O tempo relativo para voltar está próximo do tempo restante de bateria: nesse caso o agente inicia o seu caminho de retorno.
 - Ainda há tempo para explorar, dado que daquela posição, o retorno para base é menor do que o tempo restante de bateria: segue para o último caso da deliberação.
- Se nenhum outro caso impedir o agente, ele irá buscar a posição não visitada mais próxima e continuar explorando.

2. Resultados e análise

2.1. Resultados

Repositório com o código utilizado no trabalho

Utilizando como referência os cenários dos datasets 132v 100x80 e o 300v 90x90, temos os seguintes resultados:

Se tratando do dataset 132v 100x80, temos um total de 108 vítimas encontradas e um mapa com 6153 posições. Esse número baixo de vítimas e o mapa quase completo é devido à pouca bateria disponibilizada para os exploradores, dificultando a análise e o trabalho dos mesmos.

Quando falamos do dataset 300v 90x90, temos um resultado muito melhor, com 299 vítimas encontradas e um mapa com 7581 posições sendo feito após a junção dos mapas dos exploradores.

2.2. Cluster

Na questão da clusterização, utilizamos o K-Means para realizar a clusterização e obtivemos os seguintes resultados:

No dataset 132v 100x80, foram criados 4 clusters, onde tiveram o número de vítima sendo igual a 15, 28, 29, 36 nos clusters. Podemos ver como o K-Means fez a clusterização das vítimas, separando em 4 clusters como foi pedido nas orientações do projeto.

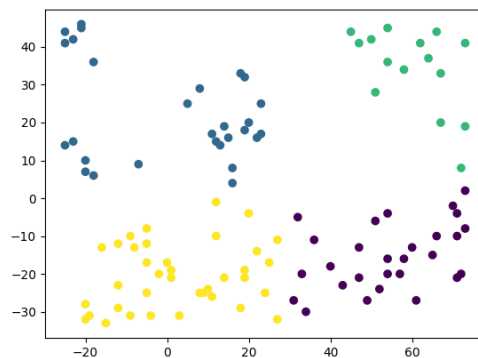


Figura 2. Gráfico do cluster 132v

Neste caso ele priorizou algumas áreas com mais vítimas e outra com menos, separando em 2 clusters bem distantes possuindo respectivamente 36 e 15 vítimas.

No dataset 300v 90x90, foram criados 4 clusters, onde tiveram o número de vítima sendo igual a 73, 76, 76 e 78 nos clusters. Podemos ver como o K-Means fez a clusterização das vítimas, separando em 4 clusters como foi pedido nas orientações do projeto.

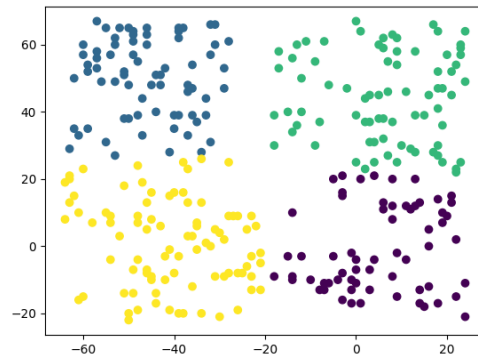


Figura 3. Gráfico do cluster 300v

Neste caso podemos ver que a separação dos clusters foi mais uniforme, tendo uma diferença baixa de vítimas entre os clusters.

3. Conclusões

De forma geral, a realização do trabalho foi um pouco complexa devido à falta de conhecimento sobre a utilização de agentes e à adaptabilidade ao código imposto. Leva um tempo para se acostumar e compreender profundamente o que o código propõe. O desenvolvimento do trabalho alcançou os objetivos estabelecidos pelo professor no enunciado, adaptando-se bem a diferentes cenários, especialmente aos disponibilizados no código-fonte.

Quanto a melhorias para trabalhos futuros, consideramos que uma lógica na qual os agentes não percorram posições já visitadas por outros agentes seria interessante. No entanto, um grande desafio para essa implementação é a possibilidade de os agentes encontrarem obstáculos ao consultar posições não mapeadas. Portanto, isso deve ser observado com atenção.

Quanto à solução atual, podemos considerá-la neutra no momento, já que a clusterização ainda não considera os dados vitais das vítimas, focando apenas nas suas posições.

4. Apêndice

Repositório com o código utilizado no trabalho

Para executar o código, precisa ter o Python no sistema e instalar as seguintes bibliotecas.

```
pip install pygame  
pip install numpy  
pip install scikit  
pip install pandas
```