

# Tarefa 01 - Elaborar e Coordenar Ações

Andrey Fabris<sup>1</sup>, Clovis Gomes<sup>2</sup>, João Pedro Wiedmer<sup>2</sup>

<sup>1</sup>PPGSE – Universidade Tecnológica Federal do Paraná (UTFPR)  
CEP 80230-901 – Curitiba – PR – Brazil

<sup>2</sup>PPGCA – Universidade Tecnológica Federal do Paraná (UTFPR)  
CEP 80230-901 – Curitiba – PR – Brazil

## 1. Introdução

A exploração em ambientes desconhecidos é uma tarefa importante para dispositivos autônomos [Feng et al. 2023]. Neste contexto, algoritmos de busca como o *Depth-First Search* (DFS) e o A-estrela podem ser empregados para explorar e percorrer tais ambientes [Krueger et al. 2011]. Este trabalho compreende a implementação de processos de exploração de ambiente, busca e clusterização de vítimas num ambiente simulado [Tacla 2024]. Os resultados obtidos são comparados com estratégias *baseline* randomizadas para avaliar a eficiência dos algoritmos utilizados.

## 2. Metodologia

Os ambientes de simulação de exploração e resgate foram obtidos de [Tacla 2024]. Do código original, desenvolvido em *python*, foi modificado o algoritmo de busca. Além disso, implementou-se um algoritmo de *clusterização*. As subseções seguintes detalham melhor a metodologia empregada.

### 2.1. Caracterização do problema e dos ambientes

O problema em questão tem como objetivo encontrar e separar em clusters as vítimas distribuídas no mapa do simulador. Para isso, quatro agentes são lançados no ambiente, os quais não possuem quaisquer informações prévias a respeito do mapa e das vítimas. Portanto, trata-se de um problema de busca cega, multiagente, cooperativo, determinístico, estático e discreto.

De acordo com a Tabela 1, dois ambientes distintos foram elencados para realização de testes e análise dos resultados. Além das características expostas, eles também diferem em posições de base, obstáculos e dificuldade de cada posição.

**Tabela 1. Caracterização dos ambientes de teste**

Ambiente	Tamanho	Nº de Vítimas	Nº Agentes	Tempo de Bateria
Ambiente 1	100x80	132	4	2500
Ambiente 2	100x80	225	4	2000

### 2.2. Estratégias de busca

A estratégia de busca em profundidade *online* (online-DFS) foi implementada devido à capacidade de um agente explorar rapidamente em profundidade antes de retroceder, o que é ideal para ambientes desconhecidos. A simplicidade e a rapidez da DFS são vantajosas

para a exploração cega [Putri et al. 2011]. Além disso, foi adicionada uma verificação de posição que evita a revisitação de coordenadas. Se todas as posições adjacentes já tiverem sido visitadas, o agente retorna até encontrar uma coordenada não explorada. Cada explorador seguiu uma sequência única de ações de exploração, com as possibilidades de ir para todas as posições ortogonais e diagonais.

A fim de maximizar o tempo de exploração, os agentes calculam o tempo e o caminho de retorno periodicamente. Quando este for ligeiramente maior que o tempo restante, inicia-se o retorno à base. O cálculo do tempo de retorno é feito por meio do algoritmo A\*, que permite uma busca heurística para resolver o caminho mais curto para um objetivo em um mapa estático e com conhecimento de informações [Bao et al. 2022]. Logo, cada explorador computa o caminho mais curto de retorno, ao considerar o próprio mapa explorado e a localização da base. As funções de custo  $g$  e heurística  $h$  adotadas são, respectivamente, o custo de acesso para próxima posição e a distância até seu objetivo, calculada por  $x^2 + y^2$ , onde  $x$  representa a distância no eixo x, e  $y$  a distância no eixo y.

### 2.3. Divisão de Vítimas

Após o retorno dos exploradores, o mapa e o conjunto de vítimas encontradas por cada explorador são agrupados e enviados ao líder dos agentes de resgate, que realiza a divisão de vítimas em grupos. Cada grupo é delegado a um agente de resgate. Para tanto, implementou-se um algoritmo de *Clusterização K-means* com 4 *clusters*. Com grande flexibilidade, eficiência, simplicidade e baixa complexidade computacional, o *K-means* têm uma ampla aceitação para os mais diferentes problemas de agrupamento [Ikotun et al. 2023]. A divisão é feita somente com base nas coordenadas ( $x$ ,  $y$ ) das vítimas, pois pretende-se enviar cada agente de resgate a uma região do mapa.

## 3. Simulações e Análises

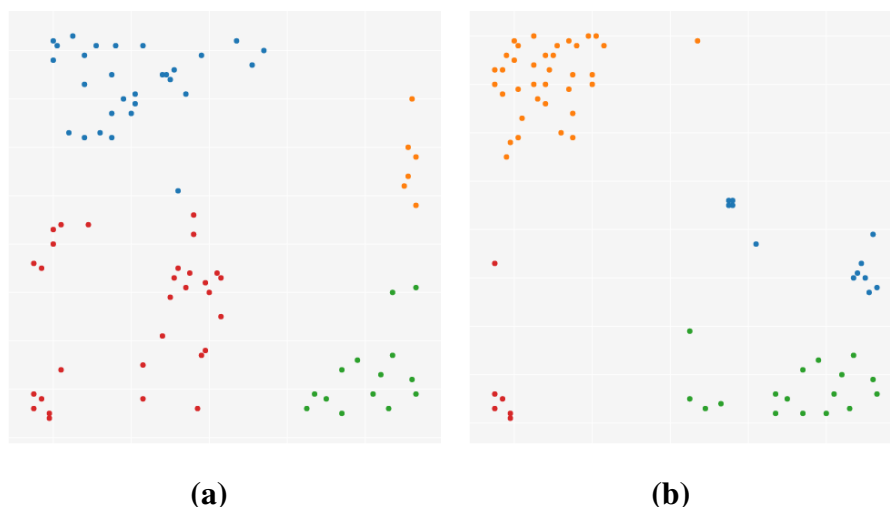
Esta seção contém os resultados obtidos pela estratégia de busca e *clusterização* adotada nos ambientes de testes. Os resultados da exploração são comparados com a média de 10 simulações do agente randômico utilizado por [Tacla 2024] nos mesmos cenários, conforme a Tabela 2. Outros resultados podem ser verificados no Apêndice A.

**Tabela 2. Comparação de Buscas**

Ambiente	Tipo de Busca	Nº Agentes	Vítimas Encontradas	$V_e$	$V_{eg}$
Ambiente 1	DFS	4	79	0,59	0,62
	Randômico	1	4,33	0,03	0,03
Ambiente 2	DFS	4	72	0,32	0,33
	Randômico	1	5,9	0,03	0,03

A comparação entre os métodos DFS e randômico revela diferenças significativas entre as abordagens. O método DFS demonstra uma capacidade superior de explorar extensas áreas do mapa, resultando na localização de um maior número de vítimas ( $V_e$ ) e um aumento no peso por severidade das vítimas encontradas ( $V_{eg}$ ). Esta superioridade é evidente mesmo quando se considera a variação no número de agentes empregados, pois o aumento de  $V_e$  e  $V_{eg}$  é proporcionalmente maior ao incremento do número de agentes. Estes resultados mostram que as ações coordenadas do DFS são mais eficazes na exploração que as movimentações randômicas do outro método.

A Figura 3 exibe os *clusters* de vítimas definidos para os ambientes 1 e 2. Cada cor representa um *cluster*.



**Figura 1. Clusters das vítimas dos ambientes 1 (a) e 2 (b)**

Percebe-se que cada *cluster* ocupa, aproximadamente, um quadrante do mapa 100x80, especialmente em 1(a). Isso é condizente com a utilização das coordenadas como variáveis de entrada no processo, e favorece a delegação dos agentes de resgate para zonas específicas. Por outro lado, os *clusters* apresentam grande variação no número de vítimas, o que se configura num contra para o resgate.

**Tabela 3. Métricas de clusterização**

Ambiente	SSE	Coef. Silhueta
Ambiente 1	24965	0,53
Ambiente 2	14182	0,62

A Tabela 3 apresenta os SSE e os coeficientes de silhueta dos ambientes simulados. Estas métricas sugerem uma *clusterização* razoável em ambos os casos, pois os coeficientes de silhueta foram inferiores a 0,7. Nota-se também que os *clusters* obtidos no Ambiente 2 são mais adequados que os obtidos no Ambiente 1, por conta da menor SSE e do maior coeficiente de silhueta.

#### 4. Conclusões

Com base na análise dos resultados obtidos, observa-se que o algoritmo proposto apresenta uma generalização eficaz na localização de vítimas, visto que teve performance satisfatória nos dois ambientes estudados. Por outro lado, a estratégia de retorno à base utilizando o algoritmo A\* poderia ser aprimorada calculando-se o caminho ótimo de retorno em todas as iterações, a fim de maximizar o tempo de exploração. Em termos éticos, a *clusterização* de vítimas com base na localização pode negligenciar aquelas em condições mais críticas, visto que não leva em conta o estado de saúde. A abordagem ideal deve priorizar as vítimas mais graves. Contudo, para tanto, é necessário dispor de modelos que relacionem os sinais vitais ao estado de saúde das vítimas. Trabalhos futuros podem obter modelos dessa natureza através de outras técnicas de *Machine Learning*.

## Referências

- Bao, W., Li, J., Pan, Z., and Yu, R. (2022). Improved a-star algorithm for mobile robot path planning based on sixteen-direction search. In *2022 China Automation Congress (CAC)*, pages 1332–1336.
- Feng, A., Xie, Y., Sun, Y., Wang, X., Jiang, B., and Xiao, J. (2023). Efficient autonomous exploration and mapping in unknown environments. *Sensors*, 23(10).
- Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., and Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210.
- Krueger, R., Simonet, G., and Berry, A. (2011). A general label search to investigate classical graph search algorithms. *Discrete Applied Mathematics*, 159(2):128–142.
- Putri, S. E., Tulus, T., and Napitupulu, N. (2011). Implementation and analysis of depth-first search (dfs) algorithm for finding the longest path.
- Tacla, C. A. (2024). <https://github.com/tacla/victimsim2>. Acesso em 13/03/2024.

## A. Resultados Adicionais

Esse apêndice contém tabelas e imagens de resultados de exploração e clusterização nos demais ambientes de testes.

### A.1. Resultados de Exploração

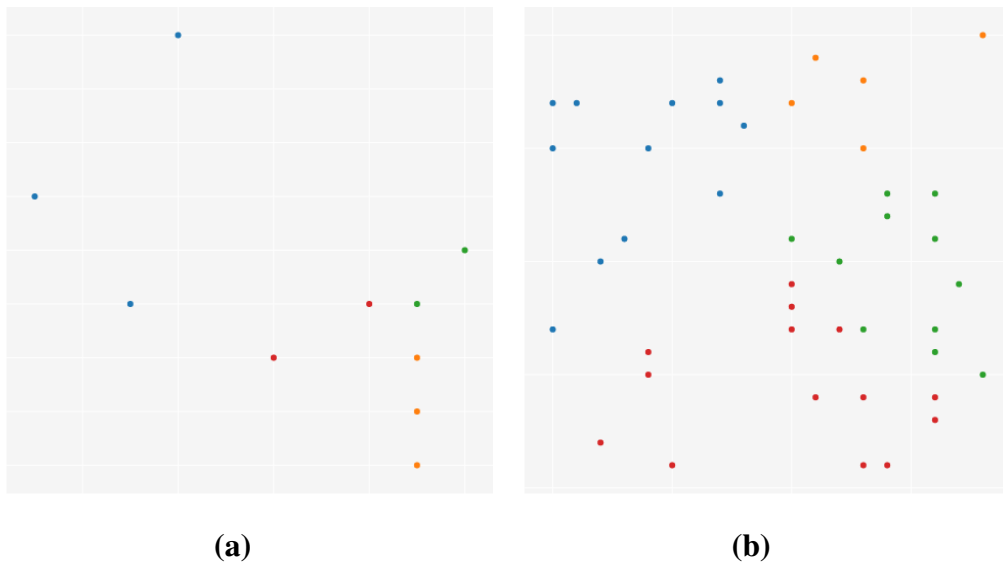
A Tabela 4 demonstra os resultados obtidos com quatro agentes de exploração para os quatro diferentes mapas disponibilizados por [Tacla 2024].

**Tabela 4. Comparativos de resultados de exploração.**

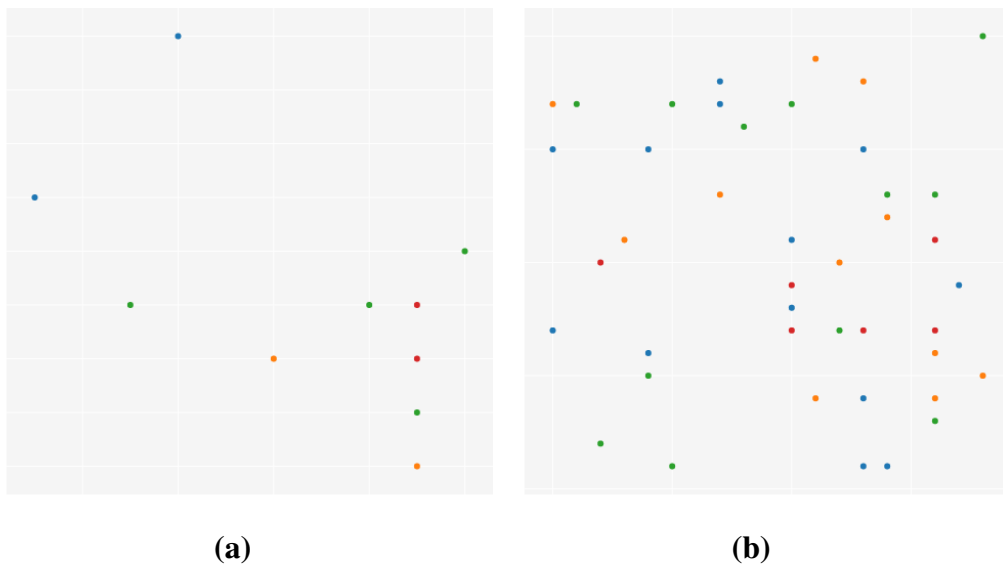
Nome Mapa	$V_{e1}$	$V_{e2}$	$V_{e3}$	$V_{e4}$	$V_{eg}$
data_10v_12x12	3 (100%)	3 (100%)	2 (100%)	2 (100%)	1.00
data_42v_20x20	5 (100%)	21 (100%)	8 (100%)	8 (100%)	1.00
data_132v_100x80	13 (72%)	45 (48%)	19 (59%)	2 (50%)	0.62
data_225v_100x80	12 (35%)	48 (34%)	11 (23%)	0 (0%)	0.33

### A.2. Resultados de Clusters

A Figura 3 mostra os cluster de vítimas para os mapas "data\_10v\_12x12" e "data\_42v\_20x20". Já Figura 2 demonstra a diferença de clusterização utilizar as características de posição e sinais vitais para o agrupamento realizado pelo *k-means* para o mapa "data\_42v\_20x20". Cada ponto refere-se a posição da vítima no mapa e sua cor corresponde ao seu agrupamento.



**Figura 2. Clusters das vítimas considerando somente as coordenadas. Ambientes 12x12 (a) e 20x20 (b)**



**Figura 3. Clusters das vítimas considerando somente os sinais vitais. Ambientes 12x12 (a) e 20x20 (b)**

## B. Instruções para a Simulação

A simulação é iniciada ao compilar o arquivo `main.py`, onde deve-se colocar o caminho do mapa a ser testado na variável `"data_folder_name"`. Os agentes devem possuir os seguintes nomes dentro de seus arquivos de configuração:

- Explorador 1: NAME EXPLORER1BLUE
- Explorador 2: NAME EXPLORER2GREEN
- Explorador 3: NAME EXPLORER3PURPLE
- Explorador 4: NAME EXPLORER4RED
- Resgate 1: NAME RESCUER1PINK

- Resgate 1: NAME RESCUER2CYAN
- Resgate 1: NAME RESCUER3YELLOW
- Resgate 1: NAME RESCUER4ORANGE

Além das biblioteca utilizadas no VictimSim2 de [Tacla 2024], são necessárias as seguintes bibliotecas instaladas no ambiente do *python*:

- numpy
- scikit-learn
- math
- warnings
- heapq