Inteligência Artificial: Relatório da implementação de algorítmos de busca

Nadyan Suriel Pscheidt[1]

[1]Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina
Centro de Ciências Tecnológicas – Joinville – SC – Brasil

nadyan.suriel@gmail.com

1. Introdução

Formas de buscar um resultado possuem uma ampla história na área de otimização de algoritmos. De acordo com [Whitley et al. 1996], inúmeros estudos foram realizados na tentativa de mostrar a efetividade de algoritmos de busca de forma experimental e também empírica. Porém, essas comparações são muito relativas quando levado em consideração o problema a ser abordado por tais algoritmos, pois determinados algoritmos possuem bons desempenhos em certos problemas, enquanto demonstram baixo desempenho em outros, enquanto outros algoritmos fazem o oposto. Com isso, é possível verificar que esses algoritmos são projetados e configurados para problemas em particular.

No trabalho aqui apresentado, são descritos os algoritmos de busca *Breadth-First Search* (BFS), Busca de Custo Uniforme e A*. Para cada um, é descrita a forma de implementação assim como estratégias utilizadas. Além dessas descrições, existe um comparativo entre os algoritmos através de experimentos.

O relatório aqui apresentado é composto de quatro seções. A problemática na qual é apresentado o problema, expondo suas características. O Modelo Implementado, na qual estratégias de implementação são discutidas. A seção de Experimentos, Resultados e Análise na qual descreve o ambiente de teste, assim como os resultados e análise dos mesmos, e por fim, a Conclusão contendo as considerações finais sobre o trabalho desenvolvido e sobre os resultados obtidos e as analises.

2. Problemática

O problema abordado envolve dois elementos: O cenário (ambiente) e um robô (agente). Esse robô deve ser capaz de calcular um caminho até seu destino seguindo a lógica dos algoritmos implementados.

O ambiente proposto na qual o agente atua é um terreno composto de diversos tipos de áreas, cada área com um grau de dificuldade para a movimentação do agente. Esse grau de dificuldade também pode ser chamado de Custo do Terreno. O ambiente é ilustrado na Figura 1 e Figura 2. Cada tipo de área do Terreno possui um Custo próprio. A área verde representa ser sólido e plano, tendo custo 1 e representado pelo número 0. A área com a coloração marrom representa ser montanhoso, com custo 5 e representado pelo número 1. Com custo 10 e representado pelo número 2, a área azul representa o pântano, e por fim, a área vermelha representa fogo, com custo 15 e representado pelo número 3.

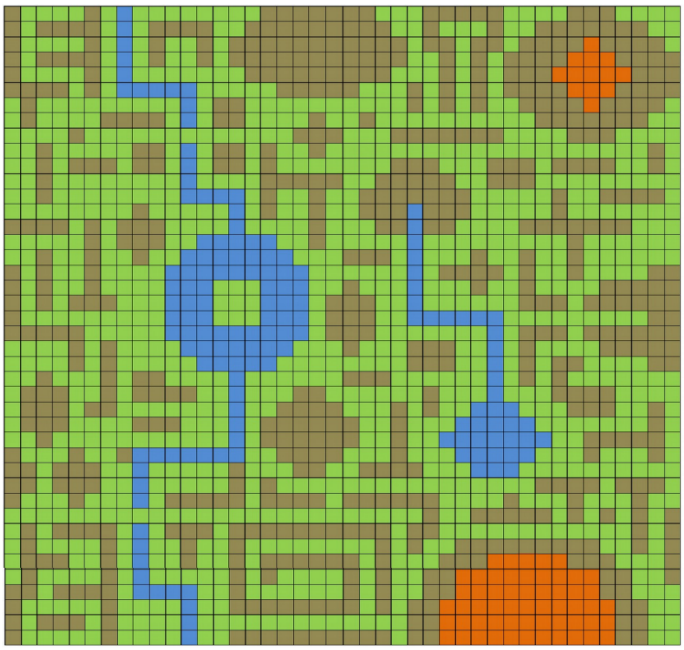

Figura 1. Ambiente: Representação conceitual

```
  0 0 0 0 1 0 2 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0
1 0 1 1 1 1 0 2 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 1 0 0 1 1 1 1 1 1 1 0 0
1 0 0 0 0 1 0 2 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 3 1 1 1 1 0
1 0 1 1 1 1 0 2 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 1 1 1 3 3 3 1 1 1 1
1 0 0 0 0 1 0 2 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 0 1 1 1 3 3 3 3 3 1 1 1
1 1 1 1 0 1 0 2 2 2 2 0 0 0 0 0 1 1 1 1 1 0 0 0 1 0 1 0 1 0 1 1 1 1 3 3 3 1 1 1 1
0 1 0 0 0 0 0 0 0 0 0 2 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 1 1 1 3 1 1 1 1 0
0 1 0 0 0 0 1 1 1 1 0 2 0 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0
0 0 1 0 0 0 0 0 1 1 0 2 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 1 0 1 1 1 1 0 1 1 0 2 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 1 0 1 1 1 1 0 0 1 0
0 0 1 0 0 0 0 0 0 0 0 0 2 0 0 0 0 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 2 2 2 2 0 1 0 0 1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 1 1 1 1 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 2 0 1 0 0 1 0 0 1 1 1 2 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0
1 1 1 1 0 1 0 1 1 1 1 0 0 0 0 2 0 0 0 0 1 0 0 0 1 1 2 1 1 0 0 0 0 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 0 1 1 1 1 0 0 2 2 2 2 2 0 0 1 0 0 0 0 0 2 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 1 0 0 1 0 0 1 0 0 2 2 2 2 2 2 2 2 0 0 0 1 1 1 0 2 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0
1 0 1 0 0 1 0 0 0 0 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 2 0 1 1 1 1 0 1 0 0 1 0 0 0 0 1 1
1 0 1 0 0 1 0 0 0 0 0 2 2 2 0 0 0 2 2 2 0 0 1 0 0 0 2 0 0 0 1 0 0 1 0 0 0 0 1 1 1 1
1 0 1 0 1 1 1 1 0 0 0 2 2 2 0 0 0 2 2 2 0 1 1 1 0 0 2 0 0 0 0 0 0 1 1 1 0 1 1 1 1 1 1
1 0 0 0 0 0 0 0 0 0 0 2 2 2 0 0 0 2 2 2 0 1 1 1 0 0 2 2 2 2 2 2 0 0 0 0 0 0 0 0 1 1
1 1 1 1 1 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 0 1 1 1 0 0 0 0 0 0 0 2 0 1 1 1 0 1 1 1 1 1
0 0 0 0 1 0 1 0 0 0 0 0 2 2 2 2 2 2 2 2 0 0 0 1 0 0 0 1 1 1 0 0 2 0 1 0 0 0 0 1 1 1
0 0 0 0 1 0 1 0 0 0 0 0 0 2 2 2 2 2 2 2 0 0 0 0 0 0 0 1 1 1 0 0 2 0 1 1 0 0 0 0 1 1
0 0 1 0 0 0 1 0 1 1 0 0 0 0 2 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 2 0 1 0 1 0 0 0 0 0 0
0 1 1 1 0 0 1 0 1 1 1 1 0 0 2 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 2 0 0 0 1 1 1 1 0 0 0
0 1 1 1 0 1 1 0 0 0 0 0 0 2 0 0 0 1 1 1 0 0 0 1 0 1 0 0 2 2 2 2 2 0 0 0 1 1 1 0 0 1 0
0 1 1 1 0 0 0 0 1 1 1 0 0 0 2 0 1 1 1 1 1 1 0 0 1 0 0 0 2 2 2 2 2 0 0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 2 0 1 1 1 1 1 1 1 0 0 1 0 0 0 2 2 2 2 2 0 0 0 1 1 1 0 1 0
1 0 0 0 1 0 0 1 2 2 2 2 2 2 2 0 0 1 1 1 1 1 0 0 0 1 0 0 0 2 2 2 2 2 0 0 0 1 0 0 0 1 0
1 1 0 1 1 1 0 0 2 0 0 0 0 0 1 0 0 0 1 1 0 0 1 1 1 1 0 0 0 2 2 2 0 0 0 0 0 0 0 0 0 0
1 1 0 1 1 1 0 0 2 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 0
1 0 0 0 1 0 0 0 2 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 1 0 1 0 1 0 1
0 1 0 1 1 1 1 1 0 2 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1
0 1 0 0 0 0 0 1 0 2 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 1
0 1 1 1 0 0 1 0 2 0 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 3 3 3 3 3 3 3 3 3 1 1 0 0 1
0 1 0 0 0 0 1 0 2 0 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 1 1 3 3 3 3 3 3 3 3 3 3 3 1 1 0 0 1
1 1 0 1 1 1 1 0 2 2 2 2 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 1 3 3 3 3 3 3 3 3 3 3 3 1 1 0 0 0
1 0 0 0 0 0 1 0 0 0 0 0 2 0 0 1 1 1 1 1 1 1 0 1 0 0 1 1 3 3 3 3 3 3 3 3 3 3 3 1 1 0 0 0
1 0 1 1 1 1 0 1 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 3 3 3 3 3 3 3 3 3 3 3 1 1 0 0
1 0 0 0 0 0 1 0 0 0 0 2 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 3 3 3 3 3 3 3 3 3 3 3 3 1 1 0
```

Figura 2. Ambiente: Representação computacional

A seguir, uma análise das características referentes ao agente:

- **Agente:** Robô de Busca;
- **Medida de desempenho:** Custo do caminho encontrado (minimizar);
- **Ambiente:** Terreno com áreas de custos diferentes, o próprio robô e os pontos de origem e destino;

- **Sensores:** Perceber o custo do terreno, posição e possíveis direções; e
- **Atuadores:** Movimentação.

Assim como a análise do agente, existe a análise das características do ambiente:

- **Ambiente:** Terreno com áreas de custos diferentes;
- **Parcialmente Observável:** pois o robô só consegue analisar as áreas adjacentes;
- **Determinístico:** Pois é possível prever o que acontecerá a cada iteração pela lógica do algoritmo em execução;
- **Sequencial:** Pois um caminho escolhido irá afetar o custo final;
- **Estático:** Pois o ambiente não muda enquanto o agente está atuando, ou seja, os custos das áreas não são modificados;
- **Discreto:** Pois as variáveis envolvidas são discretas, como tamanho do terreno, custo de cada área e quantidade de nós expandidos;
- **Agente único:** Sendo ele o robô de busca, apenas.

Após a expansão dos nós seguindo a lógica do algoritmo sendo executado, a função de busca deve retornar o caminho encontrado pelo sistema. Esse caminho resposta é exibido na tela através da Representação Computacional do ambiente, como ilustrado na Figura 3.

```
2  2  0  0  0  0  0  0  2  0  1  1  1  1  0
2  2  0  0  1  0  0  0  2  0  0  0  1  0  0
2  2  0  1  1  1  0  0  2  0  0  0  0  0  0
2  2  0. In 1  1  0  0  2  2  2  2  2  2  0
2  2  0. 1  1  1  0  0  0  0  0  0  0  2  0
2  0  0. 0. 1  0  0  0  1  1  1  0  0  2  0
0  0  0  0. 0. 0. 0. 0. 1  1  1  0  0  2  0
0  0  0  0  1  1  1  0. 0. 0  0  0  0  2  0
0  1  1  0  0  0  0  0  0. 1  0  0  0  2  0
1  1  1  1  0  0  0  1  0. 1  0  0  2  2  2
1  1  1  1  1  0  0  1  0. 0. 0  2  2  2  2
1  1  1  1  1  0  0  1  0  0. 2  2  2  2  2
1  1  1  1  0  0  0  1  0  0. 0. 2  2  2  2
0  1  1  0  0  1  1  1  1  0  0. 0. 2  Fi 2
0  0  0  0  0  0  0  0  0  0  0  0. 0. 0. 0
1  1  1  0  1  1  1  1  1  1  0  0  0  0  1
```

Figura 3. Caminho calculado pelo algoritmo A*. (Ilustração com o ambiente ampliado)

3. Modelo Implementado

3.1. Estrutura

O modelo foi implementado utilizando a linguagem de programação C, com algumas bibliotecas do C++, como a biblioteca de lista encadeada. A estrutura do sistema consiste em três algoritmos de busca distintos [Parpinelli 2017a, Parpinelli 2017b]:

- ***Breadth-First Search* (BFS):** Na qual constrói uma árvore de estados a partir do estado inicial, aplicando todas as regras possíveis aos estados do nível mais baixo. Em outras palavras, é uma busca em árvore na qual começa pelo nó raiz e em seguida explora todos os seus vizinhos. Também é chamada de Busca em Amplitude. Esse sempre encontrará uma solução caso ela exista, e essa solução será a primeira encontrada na menor profundidade da árvore;
- ***Busca de Custo Uniforme*:** Diferente da busca BFS, o algoritmo de Busca Uniforme expande o nó da fronteira que apresenta o menor custo, e não todos os seus vizinhos. Esse algoritmo é completo e ótimo; e
- ***Busca A*:*** Apresenta uma combinação do algoritmo BFS com o algoritmo de Custo Uniforme. O algoritmo A* irá expandir o nó da fronteira que apresenta o menor valor de f(n), sendo:
  - $n =$ nó na fronteira;
  - $f(n) = g(n) + h(n)$;
    - $g(n) =$ distância de **n** ao nó inicial; e
    - $h(n) =$ distância estimada de **n** ao nó final.

  A busca A* é completa e ótima.

A distância implementada no sistema é denominada **Distância de Manhattan**, na qual calcula a distância entre dois pontos de dados que seguem a estrutura de uma grade, ilustrada na Figura 4.

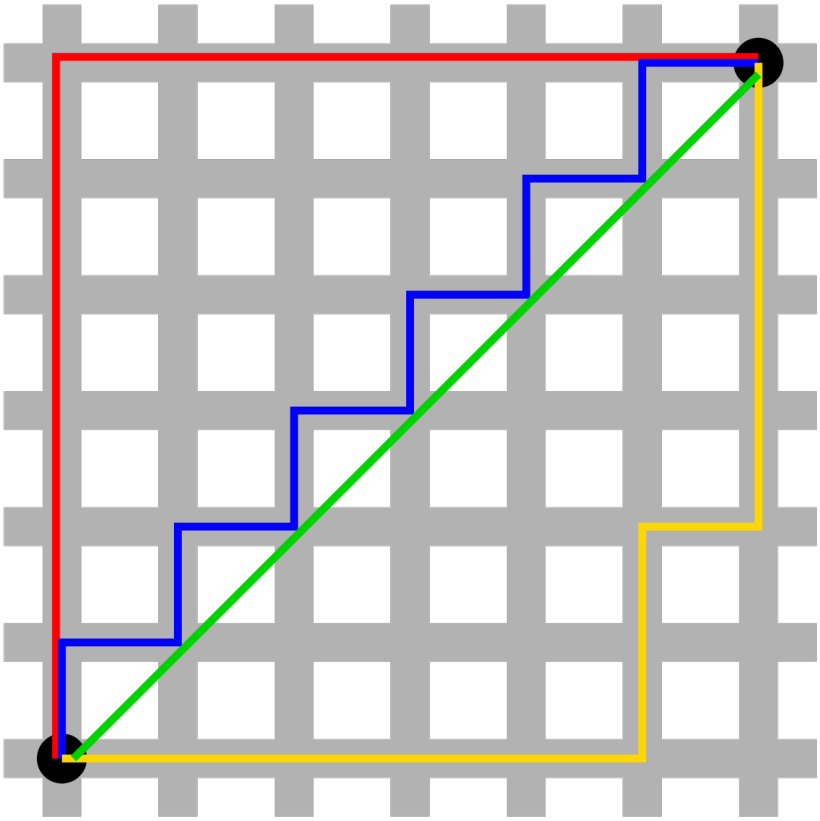

Figura 4. Representação das distâncias

Na Figura 4, a linha reta linear no centro representa a distância euclidiana clássica,

enquanto a linha alternada no centro representa a distância de *Manhattan*, seguindo a estrutura de grade.

3.2. BFS

O modelo implementado do algoritmo de busca BFS utilizou a estrutura de dados em lista para realizar as interações com os nós da árvore de expansão. O nó da cabeça da lista é expandido e seus vizinhos são adicionados na cauda da lista seguindo a ordem horária, ou seja, cima, direita, baixo e esquerda. Essas direções são coordenadas representadas na matriz terreno (na qual **x** representa linhas e **y** representa colunas) da forma: $(x-1, y)$, $(x, y+1)$, $(x+1, y)$ e $(x, y-1)$.

Como descrito anteriormente, o algoritmo BFS primeiro expande todos os nós vizinhos **sem levar em consideração o custo do terreno desses vizinhos**. Desta forma, a característica de execução desse algoritmo é ilustrada na Figura 5, na qual o nó expandido apresenta o caractere '.' ao lado do custo da área.

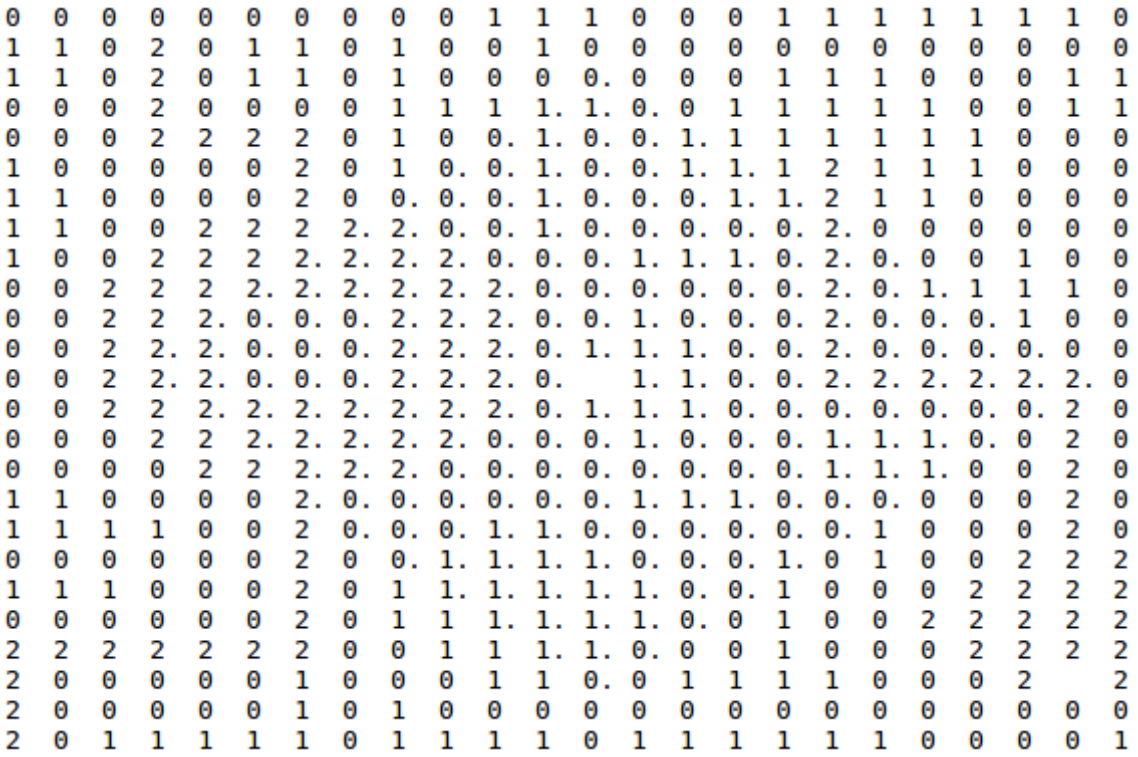

Figura 5. Procedimento de expansão de nós do algoritmo BFS (Ambiente ampliado)

Observando a Figura 5, é possível verificar o padrão de expansão de nós que se assemelha a um losango ao redor do ponto de origem (que é representado pelo espaço vazio no centro da expansão). Vale ressaltar que a Figura 5 apresenta uma visão ampliada do ambiente, com finalidade de observar os nós expandidos.

3.3. Custo Uniforme

O modelo implementado do algoritmo de Custo Uniforme armazena as distâncias dos caminhos encontrados em uma matriz denominada "distância", possuindo o mesmo ta-

manho do ambiente (42x42). Essa matriz irá armazenar a distância para cada área do terreno em relação ao ponto de origem da busca.

Inicialmente a matriz de distâncias possui o valor INFINITO[1] para cada célula. Esse valor é atualizado toda vez que um caminho de **menor custo** é encontrado para aquele determinado nó, como ilustrado no pseudocódigo:

if $distancia(nó\_vizinho) > distancia(nó\_atual) + custo(nó\_vizinho)$ **then**
    $atualiza\_distância(nó\_vizinho)$
else
    $manter\_distancia(nó\_vizinho)$
    $analisa\_nó(outro\_vizinho\_sequencia\_horária)$
end if

A expansão de nós do algoritmo de Custo Uniforme pode ser visualizada na Figura 6.

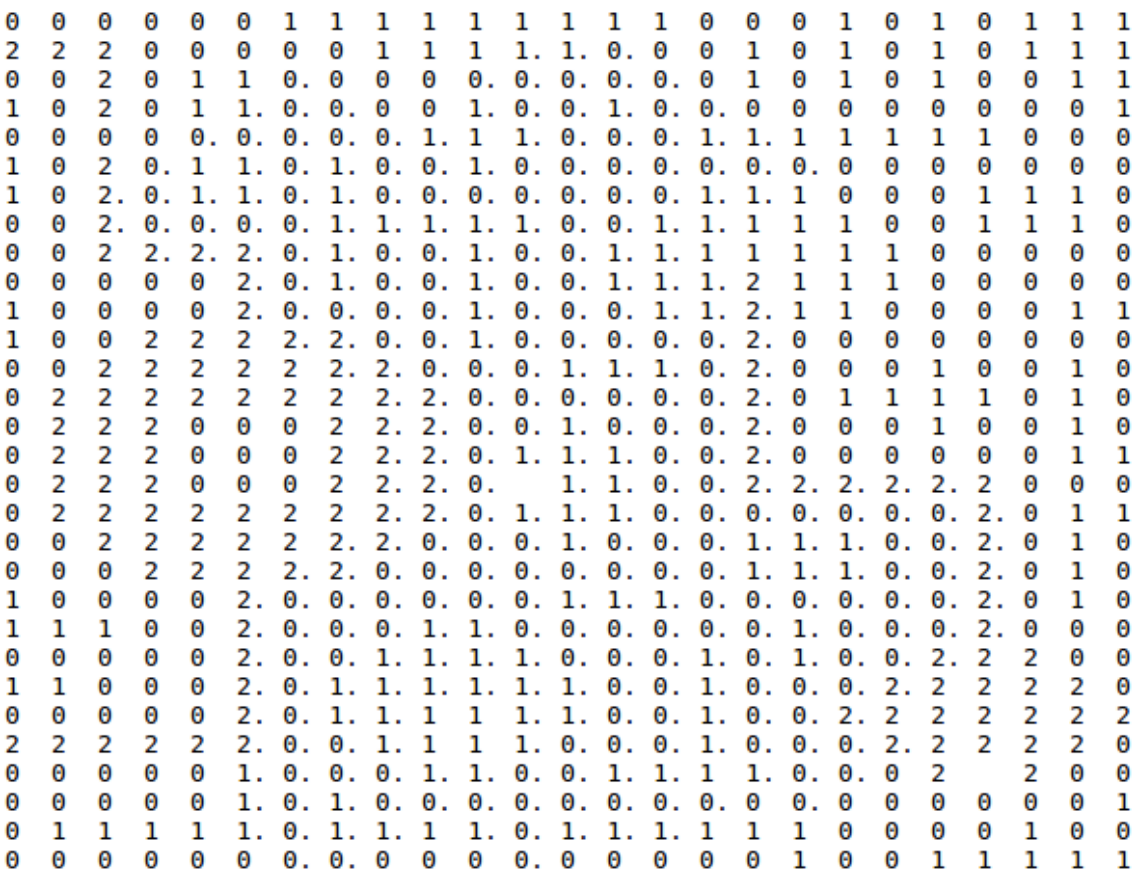

Figura 6. Procedimento de expansão de nós do algoritmo de Custo Uniforme com ambiente ampliado

Como pode ser observado, esse algoritmo expande os nós das áreas de menor custo primeiro. Ilustrado na imagem, algumas áreas representadas pelo número 2 nas quais não

[1] O valor 99999 é definido como INFINITO por questões denotacionais

foram expandidas por conta do seu custo superior aos vizinhos 1 ou 0.

3.4. A*

Semelhante ao algoritmo de Custo Uniforme, o algoritmo A* utiliza uma matriz de distâncias juntamente com uma matriz de distancias estimadas, que armazena as distâncias totais estimadas de cada nó até o ponto de destino.

As distâncias do algoritmo A* são calculadas com o método da Distância de *Manhattan*, que fornece a seguinte fórmula [ImprovedOutcomesSoftware 2004]:

$$manhattan(atual, destino) = |x\_atual - x\_destino| + |y\_atual - y\_destino|$$

Assim como a Busca Uniforme, existe a matriz de distâncias que é atualizada se encontrar um caminho de custo inferior. A ordem de busca segue o sentido horário, como no BFS e no Custo Uniforme, como descrito no pseudocódigo:

if $distancia\_estimada(nó\_vizinho) > distancia(nó\_atual) + custo(nó\_vizinho) + manhattan(nó\_vizinho, destino)$ **then**
    $atualiza\_distância\_estimada(nó\_vizinho) + manhattan(nó\_vizinho, destino)$
    $atualiza\_distancia(nó\_vizinho)$
else
    $manter\_distancia\_estimada(nó\_vizinho)$
    $analisa\_nó(outro\_vizinho\_sequencia\_horária)$
end if

A expansão de nós do algoritmo A* pode ser visualizada na Figura 7.

```
0  0  0  1  1  1  1  1  0  0  1  1  1  1  1  0  0  1  1  1  0  0
2  2  0  1  0  0  1  0  0  1  1  1  1  1  1  1  0  0  0  0  0  1
0  2  0  1  0  0  1  0  0  1  1  1  2  1  1  1  0  0  0  0  0  1
0  2  0  0  0  0  1  0. 0  0  1  1  2  1  1  0  0  0  0  1  1  1
2  2  2  2  0  0  1. 0. 0. 0  0  0. 2  0  0  0  0  0  0  0  0  0
2  2  2  2  2  0. 0. 0. 1. 1. 1. 0. 2. 0  0  0  1  0  0  1  0  0
2  2  2  2  2  2. 0. 0. 0. 0. 0. 0. 2. 0  1  1  1  1  0  1  0  0
0  0  0  2  2  2. 0. 0. 1. 0. 0. 0. 2. 0  0  0  1  0  0  1  0  0
0  0  0  2  2  2. 0. 1. 1. 1. 0. 0. 2. 0  0  0  0  0  0  1  1  1
0  0  0  2  2  2. 0.    1. 1. 0. 0. 2. 2. 2. 2. 2  0  0  0  0
2  2  2  2  2  2. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 2. 0  1  1  1
2  2  2  2  2. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0. 2. 0  1  0  0
2  2  2. 2. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 2. 0  1  0  1
0  2. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 2. 0. 1  0  1
0  2. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 2. 0. 0  0  1
0  2. 0. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 2. 2. 2  0  0  0
0  2. 0. 1. 1. 1  1. 1. 1. 0. 0. 1. 0. 0. 0. 2. 2. 2  2  2  0  0
0  2. 0. 1. 1  1  1  1. 1. 0. 0. 1. 0. 0. 2. 2. 2  2  2  2  2  0
2  2. 0. 0. 1. 1  1. 1. 0. 0. 0. 1. 0. 0. 0. 2. 2. 2  2  2. 0. 0
0  1. 0. 0. 0. 1. 1. 0. 0. 1. 1. 1. 1. 0. 0. 0. 2.    2. 0. 0. 0.
0  1  0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1
1  1  0  1  1  1  1. 0. 1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0
0  0  0  0  0  0  0  0. 0  0  0  0  0  1. 0. 0. 1. 1. 1  1. 1  0
0  1  1  1  1  1  1  1  1  1  1  0  1  1. 0. 0. 0. 0. 0. 0  0  0
0  1  0  0  0  0  0  0  0  0  1  0  1  0  0. 1. 1. 1. 1  1  1  1
```

Figura 7. Procedimento de expansão de nós do algoritmo A* com ambiente ampliado

Como pode ser observado, esse algoritmo leva em consideração tanto o custo da área quanto a distância da mesma até o ponto de destino, expandindo uma quantidade menor de nós quando comparado ao algoritmo de Custo Uniforme. Essa área de expansão demonstra-se direcionada ao ponto de destino (que na Figura 7 está representado por uma célula vazia no canto inferior direito do ambiente), diferente do Custo Uniforme, na qual expande nós em todas as direções a partir da origem.

4. Experimentos, resultados e análise

Para os experimentos, foram escolhidos 10 conjuntos aleatórios de pontos de origem e destino (linha, coluna):

- **Conjunto 1:** Origem: (1, 40); Destino: (40, 1);
- **Conjunto 2:** Origem: (3, 3); Destino: (31, 30);
- **Conjunto 3:** Origem: (3, 40); Destino: (40, 3);
- **Conjunto 4:** Origem: (5, 20); Destino: (20, 5);
- **Conjunto 5:** Origem: (8, 8); Destino: (8, 40);
- **Conjunto 6:** Origem: (20, 20); Destino: (25, 25);
- **Conjunto 7:** Origem: (21, 21); Destino: (40, 6);
- **Conjunto 8:** Origem: (30, 20); Destino: (20, 30);
- **Conjunto 9:** Origem: (31, 30); Destino: (20, 3);
- **Conjunto 10:** Origem: (41, 20); Destino: (1, 20);

Cada algoritmo foi executado para cada um dos conjuntos. Após as execuções, foram extraídos os valores de Custo e Quantidade de Nós Expandidos de cada algoritmo, feito a média e calculado o desvio padrão. O algoritmo denotado como A*(10) consiste no algoritmo A* na qual o valor da distância de *Manhattan* é superestimada em 10 vezes.

Os resultados podem ser observados nas Tabela 1 e Tabela 2. O símbolo $\bar{x}$ representa a média, enquanto o símbolo σ representa o desvio padrão.

Algoritmo	$\bar{x}Custo$	σ
BFS	170,5	114.8
Uniforme	52,1	27.6
A*	52,1	27.6
A*(10)	103,2	51.9

Tabela 1: Resultados de Custo obtidos nos experimentos

Algoritmo	$\bar{x}Nós$	σ
BFS	1243.6	501.6
Uniforme	1304.1	491.0
A*	570.3	503.6
A*(10)	117	61.8

Tabela 2: Resultados de quantidade de nós expandidos obtidos nos experimentos

A seguir, a Figura 8 ilustra os resultados, na qual o eixo **x** corresponde ao Custo e o eixo **y** corresponde à quantidade de nós expandidos.

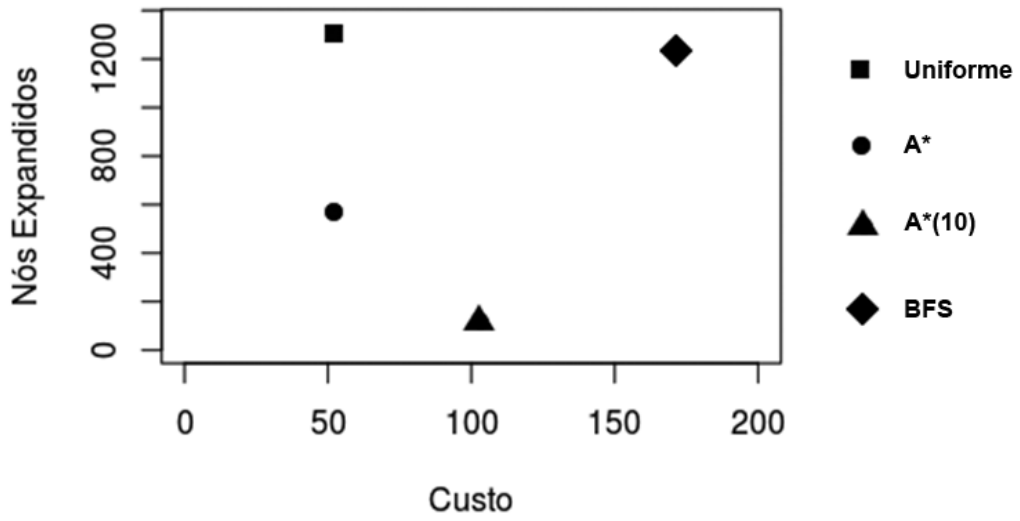

Figura 8. Resultados obtidos no experimentos

4.1. Análise dos resultados

Para analisar os resultados, introduzimos os conjuntos de Pareto no gráfico. Esses conjuntos expõe as Soluções Dominadas. Como ilustrado na Figura 9.

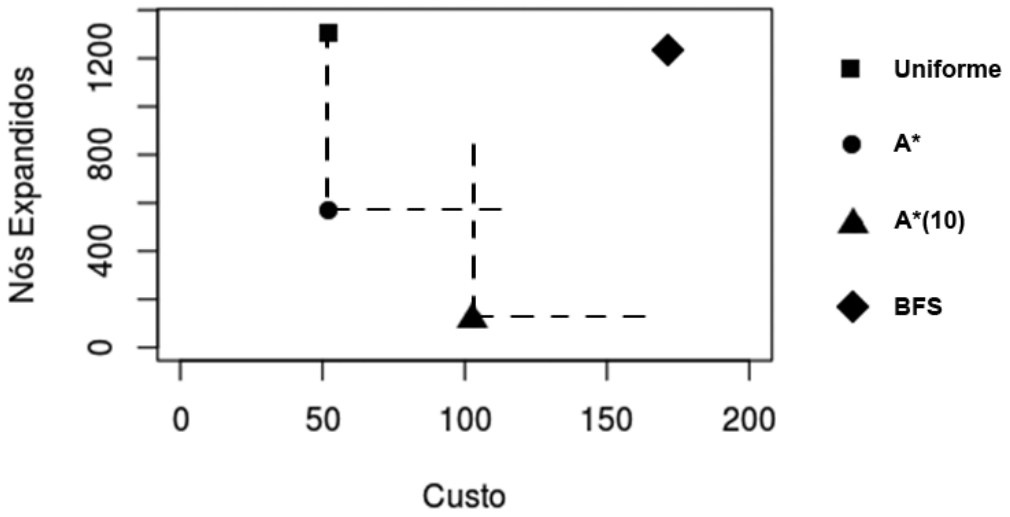

Figura 9. Gráfico de Pareto com conjunto de dominados

Observando a Figura 9, é possível observar os conjuntos de Soluções dominadas e Não Dominadas:

- **Soluções Dominadas:** Uniforme e BFS, pois de acordo com os eixos imaginários inseridos, os algoritmos A* e A*(10) estão na origem indicando que estão no ponto ótimo quando comparados com Uniforme e BFS; e

- **Soluções Não Dominadas:** A* e A*(10), por estarem no ponto de origem dos algoritmos Uniforme e BFS.

Com esses resultados, é possível afirmar que os algoritmos A* e A*(10) tiveram desempenho superior para o problema abordado quando comparados aos algoritmos de Custo Uniforme e BFS. Esse desempenho superior é tanto para a quantidade de Nós Expandidos quanto para o Custo do caminho final, embora o algoritmo A* apresenta os mesmos custos de caminhos finais do Custo Uniforme, por ambos serem algoritmos ótimos.

Outra análise que vale ser ressaltada é que o algoritmo A* quando é configurado para ter um valor de distância superestimado, como no caso A*(10), torna-se um algoritmo guloso. Enquanto quanto mais subestimado, torna-se o algoritmo de Custo Uniforme.

5. Conclusão

O relatório aqui apresentado documentou o projeto, desenvolvimento e experimentos de um sistema com o objetivo de resolver o problema de encontrar o melhor caminho em um ambiente com terreno de áreas com custos diferenciados. Esse sistema conta com a implementação de três algoritmos de busca, além da variação do algoritmo A* na qual o valor da distância é superestimado.

Com os experimentos realizados, foi possível definir a diferença entre os algoritmos nos quesitos custo do caminho final e quantidade de nós expandidos, assim como quais algoritmos obtiveram os melhores desempenhos para o problema proposto.

Referências

ImprovedOutcomesSoftware (2004). Manhattan distance metric. `http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Manhattan_Distance_Metric.htm`. Accessed: 2017-05-05.

Parpinelli, R. S. (2017a). Inteligência artificial: Busca cega. Slides para a disciplina de Inteligência Artificial.

Parpinelli, R. S. (2017b). Inteligência artificial: Busca heurística. Slides para a disciplina de Inteligência Artificial.

Whitley, D., Rana, S., Dzubera, J., and Mathias, K. E. (1996). Evaluating evolutionary algorithms. *Artificial intelligence*, 85(1):245–276.