Instituto Politécnico do Cávado e do Ave

Escola Superior de Tecnologia

Licenciatura

em

Engenharia Informática Médica

Inteligência Artificial

Bruno Rafael Mendes Oliveira – a15566

Diogo Mário Sá Fernandes – a24017

Dezembro de 2023

Esta página foi deixada em branco propositadamente.

Instituto Politécnico do Cávado e do Ave

Escola Superior de Tecnologia

Licenciatura

em

Engenharia Informática Médica

Relatório do Projeto Engenharia de Software

Gestão de Armazéns por Inteligência Artificial

Unidade Curricular

Inteligência Artificial

Nome dos Alunos

Bruno Oliveira

Diogo Fernandes

Docente da Unidade Curricular:

Profª. Joaquim Gonçalves

Dezembro de 2023

Esta página foi deixada em branco propositadamente.

Resumo

Este relatório descreve o trabalho prático realizado na UC de Inteligência Artificial, envolvendo a análise e desenvolvimento de algoritmos pesquisa num determinado armazém automático, onde este armazém tem um ponto de início onde existe um robot para coletar um produto e depois de coletar este produto o robot tem de ir para a saída. O objetivo foi aplicar os conceitos teóricos e desenvolver habilidades práticas lecionada na UC.

O trabalho prático permitiu a aplicação dos conhecimentos teóricos e destacou a importância da Inteligência Artificial bem como a implementação de vários algoritmos como a perceção de uso. Assim como consolidar os conhecimentos adquiridos ao longo da UC.

Palavras-Chaves: Inteligência Artificial, Agentes, Armazéns Automáticos, *Python*, Algoritmos de pesquisa

Abstract

This report describes the practical work carried out in the AI course, involving analysis and research development in a certain imaginary warehouse, where this warehouse has a starting point where there is a robot to collect a product and after collecting this product the robot has to go to the exit. The objective was to apply theoretical concepts and develop practical skills taught in the course. The practical work allowed the application of theoretical knowledge and highlighted the importance of Artificial Intelligence as well as the implementation of various algorithms as a perception of use. As well as consolidating the knowledge acquired throughout the course.

***Keywords**: Artificial Intelligence, Agents, Automated Warehouses, Python, Search Algorithms*

Índice

Índice de Figuras

Índice de Tabelas

Lista de siglas e acrónimos

- AI: *Artificial Inteligence*;
- UC: Unidade Curricular
- *BFS*: *Breadth-First Search*

1. Introdução

1.1. Enquadramento

No âmbito da UC de Inteligência Artificial, aprender sobre algoritmos de busca é crucial para entender como construir sistemas inteligentes. Estes algoritmos são fundamentais para a compreensão de como as máquinas podem imitar processos de raciocínio humano, especialmente em tarefas que envolvem resolução de problemas e tomada de decisões.

Os algoritmos de busca, como o Busca em Largura (*BFS*), A Estrela (A*) e *Greedy Search*, exemplificam a aplicação de conceitos de Inteligência Artificial na solução de problemas complexos de navegação e otimização. No contexto de um armazém automatizado, estes algoritmos desempenham um papel crucial na orientação de robôs para a execução eficiente de tarefas de coleta e entrega, uma aplicação prática e altamente relevante dos princípios de IA.

Este estudo contribui para o entendimento teórico de algoritmos de busca dentro do campo da Inteligência Artificial, mas também demonstra como esses conceitos são aplicados em cenários do mundo real. A análise do desempenho desses algoritmos em simulações de ambientes de armazém oferece insights valiosos sobre sua eficácia, limitações e potencial para futuras inovações na indústria de automação e logística.

Ao integrar teoria e prática, este trabalho alinha-se com os objetivos centrais da cadeira de Inteligência Artificial, proporcionando uma compreensão profunda tanto dos fundamentos teóricos quanto das aplicações práticas dos algoritmos de busca.

1.2. Objetivos

Compreender a aplicação dos algoritmos de busca em largura (*BFS*), A Estrela (A*), e *Greedy Search* em ambientes de armazém automatizado.

Analisar a eficácia destes algoritmos em diferentes cenários de simulação de armazém, com variações no tamanho da matriz e nos níveis de obstáculos.

Comparar o desempenho desses algoritmos em termos de eficiência, tempo de execução, e precisão na navegação de um robô automatizado.

Explorar as limitações e potenciais melhorias desses métodos em ambientes de armazéns reais.

2. Algoritmos

2.1. Algoritmo de Busca em Largura (*BFS*)

2.1.1. Definição

A busca em largura (*BFS*), é um algoritmo para percorrer ou pesquisar estruturas de dados em árvore ou grafo. O algoritmo começa no nó raiz (ou qualquer nó específico em um grafo) e explora todos os vizinhos desse nó antes de passar para os nós no próximo nível de profundidade. Ele visita os nós em camadas ou níveis, garantindo que todos os eles de um determinado nível sejam explorados antes de passar para o próximo. A BFS é útil para encontrar o caminho mais curto em grafos não ponderados.

2.1.2. Propriedades:

- **Ótimo para Grafos Não Ponderados:** Em grafos não ponderados, o BFS garante encontrar o caminho mais curto entre o nó inicial e qualquer outro nó no grafo, pois ele explora todos os nós igualmente em cada nível antes de passar para o próximo.
- **Completo:** O BFS é completo, o que significa que se houver uma solução, o algoritmo a encontrará. Ele explora sistematicamente todos os nós e caminhos possíveis em um grafo finito.
- **Complexidade de Espaço:** A complexidade de espaço do BFS é proporcional à largura máxima do grafo. Ele precisa armazenar todos os nós de um nível para acessar os do próximo, o que é feito através de um vetor. Portanto, em grafos grandes, o BFS pode consumir uma quantidade significativa de memória.

2.2. A Estrela

O Algoritmo A* é uma técnica de busca amplamente usada em algoritmia para encontrar o caminho mais curto entre um ponto inicial e um ponto final em um grafo.

2.2.1. Propriedades do Algoritmo A*:

- **Ótimo e Completo (com a Heurística Adequada):** O algoritmo A* é ótimo e completo se a função heurística usada for admissível, ou seja, nunca superestima o custo real para alcançar o objetivo. Isso garante que o A* encontre o caminho mais curto em um espaço de busca finito.
- **Heurística:** O coração do A* é sua função heurística, que estima o custo do caminho mais barato do nó atual para o objetivo.
- **Complexidade de Espaço:** A* mantém todos os caminhos não explorados na memória. Em espaços de busca grandes, isso pode levar a um alto consumo de memória. A complexidade de espaço é, na pior das hipóteses, exponencial em relação ao comprimento do caminho.
- **Complexidade de Tempo:** A complexidade de tempo depende fortemente da heurística. Na pior das hipóteses, pode ser exponencial, especialmente se a heurística não for eficaz. Com uma boa heurística, no entanto, A* é significativamente mais rápido do que algoritmos de busca sem informação.
- **Balanceamento entre Exploração e Eficiência:** A* equilibra a busca que se move rapidamente em direção ao objetivo, e o método de menor custo, que explora caminhos de baixo custo. Este balanceamento é alcançado através da heurística.

2.3. *Greedy Search*

Um algoritmo *Greedy Search* é uma abordagem para resolver um problema selecionando a melhor opção disponível no momento. Não se preocupa se o melhor resultado atual trará o resultado global ótimo.

2.3.1. Propriedades do Algoritmo *Greedy Search*

Não Ótimo: O *Greedy Search* não garante encontrar a solução mais ótima para um problema. Ele faz escolhas que parecem ser as melhores no momento, sem considerar o custo acumulado ou o impacto futuro dessas escolhas.

Não Completo: Este algoritmo não é completo, o que significa que não há garantia de que ele sempre encontrará uma solução, se ela existir. Em cenários com múltiplos caminhos ou obstáculos, pode falhar em encontrar um caminho até o objetivo.

Dependência da Heurística: A eficácia do *Greedy Search* é altamente dependente da qualidade da função heurística utilizada. Uma heurística apropriada pode conduzir rapidamente ao objetivo, enquanto uma heurística inadequada pode levar a resultados não atinge a mais alta qualidade ou falhas.

Eficiência em Termos de Espaço e Tempo: O *Greedy Search* é geralmente eficiente em termos de uso de memória e tempo de execução.

Aplicabilidade Prática Limitada: Embora o *Greedy Search* seja útil em alguns cenários específicos, sua aplicabilidade é limitada em problemas complexos, onde a escolha mais promissora a curto prazo não necessariamente leva ao melhor resultado global.

3. Armazéns

Neste relatório, exploramos a aplicação de algoritmos de busca em um ambiente de armazém automatizado, onde um robô é utilizado para realizar tarefas de coleta e entrega de produtos. A eficiência e eficácia do robô em navegar pelo armazém são fundamentais para otimizar as operações logísticas. Para isso, modelamos o ambiente do armazém como uma matriz, onde cada célula representa uma área específica, variando de espaços transitáveis a obstáculos.

Componentes do Armazém:

- **Robô (R):** Localizado inicialmente em uma posição específica da matriz, o robô é a unidade móvel encarregada de coletar e entregar produtos.
- **Espaço Livre (.):** Representa as áreas livres do armazém por onde o robô pode mover-se, permitindo-lhe traçar rotas para alcançar diferentes pontos.
- **Obstáculos (#):** Simulam paredes ou pilares no armazém, criando barreiras físicas. Estes obstáculos são intransponíveis para o robô, requerendo que ele planeje rotas alternativas para alcançar seu destino.
- **Produto (P):** Localização de um produto que o robô deve coletar. Este ponto é o destino inicial do robô, onde ele deve chegar para realizar a coleta.
- **Ponto de Entrega (O):** Após a coleta do produto, o robô deve levar o item até este ponto para completar a entrega.

Aplicação dos Algoritmos de Busca:

Os algoritmos de busca como DFS, BFS e A* são implementados para guiar o robô pelo armazém.

O objetivo é encontrar o caminho mais eficiente do ponto inicial (R) até o produto (P), e em seguida, do produto até o ponto de entrega (O), considerando os obstáculos presentes.

4. Desenvolvimento

4.1. Matriz para simular Armazém

A matriz serve como uma representação simplificada de um armazém, onde cada célula corresponde a uma área específica dentro do espaço físico do armazém. Esta representação é fundamental para simular e analisar o movimento do robô dentro deste ambiente.

- **Criação da Matriz:**

A matriz é inicialmente criada como uma matriz de células vazias, representadas por pontos ('.'). Cada ponto simboliza um espaço livre por onde o robô pode se mover.

- **Definição de Percentagem de Obstáculos**:

A quantidade de obstáculos é determinada por uma percentagem definida, assegurando que um certo número de células na matriz seja convertido em obstáculos.

- **Adição de Obstáculos**

Obstáculos são distribuídos aleatoriamente pela matriz para simular obstáculos no armazém. Estes são representados por '#' e indicam áreas pelas quais o robô não pode passar.

- **Posicionamento de Elementos-Chave:**

Robô (R): Uma célula é designada como o ponto de partida do robô.

Produto (P): Um local na matriz é escolhido aleatoriamente para representar onde o produto está localizado.

Ponto de Entrega (O): Um ponto é estabelecido como o destino final do robô, após a coleta do produto.

- **Aleatoriedade e Realismo:**

A posição dos obstáculos e do produto é aleatória, imitando a natureza imprevisível e variável de um armazém real.

- **Processo de Geração:**

A matriz é gerada automaticamente com estes critérios, utilizando algoritmos para assegurar a distribuição aleatória dos obstáculos e a colocação dos elementos-chave.

4.2. Algoritmo BFS

O Algoritmo de Busca em Largura (BFS) é utilizado para orientar o movimento de um robô em um armazém simulado. O objetivo é que o robô encontre e siga o caminho mais eficiente para coletar um produto (P) e, em seguida, entregá-lo em um ponto de saída (O), navegando por um espaço repleto de obstáculos (#).

1. **Implementação e Funcionamento:**

Uma interface gráfica mostra o armazém, com cores distintas marcando o robô, o produto, o ponto de saída, os caminhos explorados e os obstáculos. Isso proporciona uma visualização clara do ambiente e da trajetória do robô.

2. **Ponto de Partida:**

O *BFS* começa em um nó específico do grafo, que é geralmente referido como o nó raiz ou inicial. Este nó serve como o ponto de partida para a busca.

3. **Inicialização de Estruturas de Dados:**

Para rastrear a progressão da busca, o BFS utiliza um vetor. Todos os nós visitados são armazenados nesse vetor.

4. **Exploração do Primeiro Nível:**

O nó inicial é marcado como visitado e adicionado ao vetor. Em seguida, o algoritmo explora todos os vizinhos diretos desse nó.

5. **Passagem para Níveis Subsequentes:**

Uma vez que todos os vizinhos do nó inicial são explorados e adicionados ao vetor, o algoritmo remove o nó inicial da fila e passa para o próximo nó no vetor.

Este processo é repetido para explorar todos os vizinhos não visitados do nó atual e adicioná-los ao vetor.

A busca continua seguindo a ordem dos nós na fila, garantindo que os eles sejam explorados nível por nível.

6. **Continuação até Esgotar os Nós ou chegar ao Objetivo:**

O processo continua até que a vetor esteja vazio ou chegue ao objetivo.

Durante a busca, cada nó é visitado exatamente uma vez, assegurando uma procura completa e sistemática.

7. **Reconstrução do Caminho:**

Uma vez que o objetivo é alcançado, o algoritmo reconstrói o caminho do nó objetivo de volta ao nó inicial.

4.3. Algoritmo A Estrela

No cenário do armazém automatizado, o algoritmo A Estrela é utilizado para guiar um robô na coleta e entrega de produtos. O armazém é simulado por uma matriz, onde o robô deve encontrar o caminho mais eficiente até um produto (P), e depois até um ponto de saída (O), superando obstáculos representados por '#' na matriz.

1. **Implementação e Funcionamento:**

Uma interface gráfica mostra o armazém, com cores distintas marcando o robô, o produto, o ponto de saída, os caminhos explorados e os obstáculos. Isso proporciona uma visualização clara do ambiente e da trajetória do robô.

2. **Inicialização:**

O algoritmo começa com um conjunto aberto que contém apenas o nó inicial e um conjunto fechado vazio.

Cada nó mantém três valores importantes: o custo real do caminho do nó inicial até ele (g), a estimativa heurística do custo para alcançar o objetivo, onde é calculado o caminho de *Manhattan* (h) e a soma destes dois valores (f = g + h). Este custo real foi feito em unidades de tempo, onde andar no mesmo sentido ficaria 1 unidade mais cara e caso o sentido fosse mudado a deslocação seria 2 unidades.

3. **Exploração de Nós:**

Em cada etapa, o algoritmo escolhe o nó com o menor valor de f no conjunto aberto.

Este nó é removido do conjunto aberto e adicionado ao conjunto fechado.

4. **Expansão do Nó Atual:**

O algoritmo expande o nó selecionado, examinando todos os seus vizinhos.

Para cada vizinho, o algoritmo calcula o valor de g (custo do caminho do nó inicial até o vizinho), h (caminho de *Manhattan)* e atualiza o valor de f.

5. **Atualização do Conjunto Aberto:**

Se um vizinho não está nem no conjunto aberto nem no fechado, ele é adicionado ao conjunto aberto.

Se o vizinho já está em um dos conjuntos, mas o novo caminho é melhor (menor g), o caminho até esse vizinho é atualizado.

6. **Repetição do Processo:**

O algoritmo repete esse processo de seleção, expansão e atualização até que o conjunto aberto esteja vazio ou até que o objetivo seja alcançado.

7. **Verificação de Objetivo:**

Se o nó selecionado for o nó objetivo, o algoritmo reconstrói o caminho até o nó inicial e termina.

8. **Reconstrução do Caminho:**

Uma vez que o objetivo é alcançado, o algoritmo reconstrói o caminho do nó objetivo de volta ao nó inicial, seguindo os caminhos de menor custo registados.

4.4. Algoritmo *Gridy Search*

No ambiente simulado de um armazém, o algoritmo *Greedy Search* é empregue para guiar um robô na tarefa de localizar e coletar um produto, e posteriormente transportá-lo para um ponto de saída. O armazém é mapeado por uma matriz, com obstáculos, caminhos e locais de interesse claramente definidos.

1. **Implementação e Funcionamento:**

Uma interface gráfica mostra o armazém, com cores distintas marcando o robô, o produto, o ponto de saída, os caminhos explorados e os obstáculos. Isso proporciona uma visualização clara do ambiente e da trajetória do robô.

2. **Ponto de Partida:**

O algoritmo inicia a partir de um nó específico, considerado o ponto de partida da procura.

3. **Decisão Baseada em Heurística:**

Em cada passo, o algoritmo seleciona o próximo nó para explorar com base em uma função heurística. Essa função avalia qual opção parece levar mais rapidamente ao objetivo, sem preocupar-se com o caminho percorrido até então.

4. **Avanço Direcionado:**

O algoritmo avança para o nó escolhido, baseando-se unicamente que ele oferece em direção ao objetivo. Ele não reconsidera as escolhas anteriores, mantendo-se focado no caminho à frente.

5. **Procura pelo Objetivo:**

A busca continua seguindo a orientação da heurística, até que o objetivo seja encontrado ou até que fique claro que não há caminho viável para o objetivo a partir do trajeto escolhido.

6. **Conclusão da Busca:**

Uma vez que o objetivo é alcançado, o algoritmo reconstrói o caminho do nó objetivo de volta ao nó inicial, seguindo os caminhos de menor custo registados.

4.5. Interface Gráfica

Foi implementada uma parte gráfica para validar e perceber qual os caminhos que os algoritmos percorriam, desta forma, a trajetória encontrada pelos algoritmos é visível em verde, mostrando a rota do ponto inicial (azul) ao ponto final (rosa). Esta representação visual é útil para confirmar a validade e eficiência do caminho encontrado.

A eficiência dos algoritmos também é evidente na forma como a trajetória parece tomar decisões direcionais ótimas, evitando obstáculos (pretos) e minimizando a distância total percorrida.

As áreas que foram exploradas durante a busca, mas que não fazem parte do caminho final, estão marcadas em vermelho. Isso demonstra o processo de busca dos algoritmos, onde várias rotas potenciais são consideradas antes de se chegar à solução final.

A densidade de áreas vermelhas pode dar uma ideia da complexidade do espaço de busca e da eficiência do algoritmo em descartar caminhos não ótimos rapidamente.

A ausência de áreas vermelhas excessivas ao longo do caminho verde indica que os algoritmos não desperdiçaram recursos a explorar as áreas desnecessárias do espaço de busca.

Esta representação visual é extremamente útil na otimização de rotas, simulando assim aplicações do mundo real.

4.5.1. Interface Gráfica A*

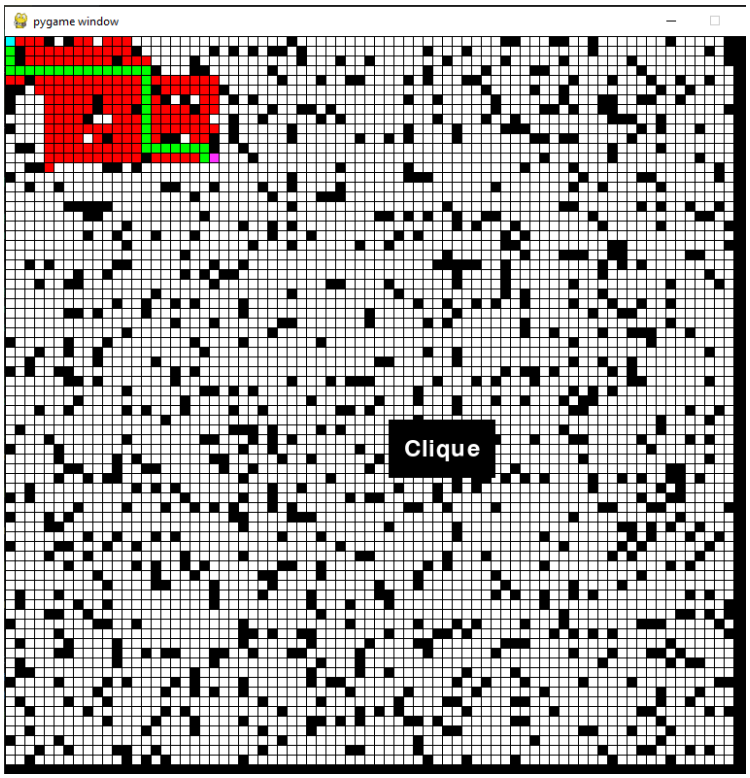


Figura 1 - Interface Gráfica A*- Robô para Produto

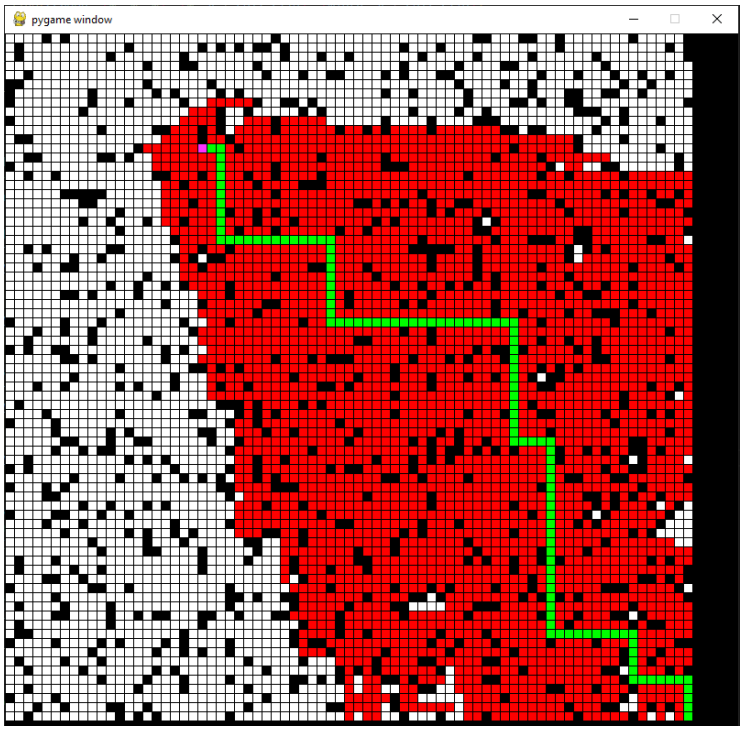


Figura 2 - Interface Gráfica A*- Produto para Output

4.5.2. Interface Greedy Search

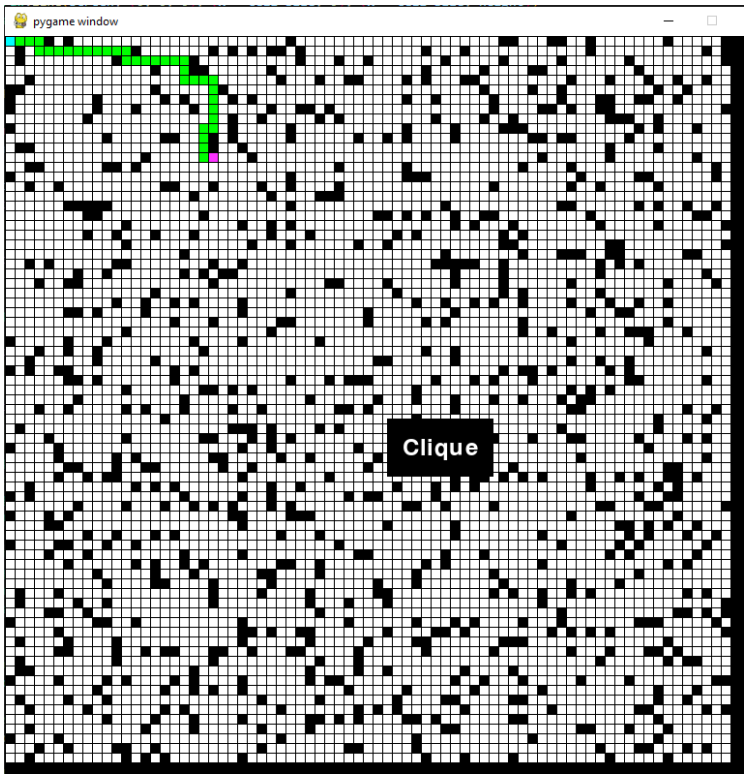


Figura 3 - Interface Gráfica *Greedy Search* *- Robô para Produto

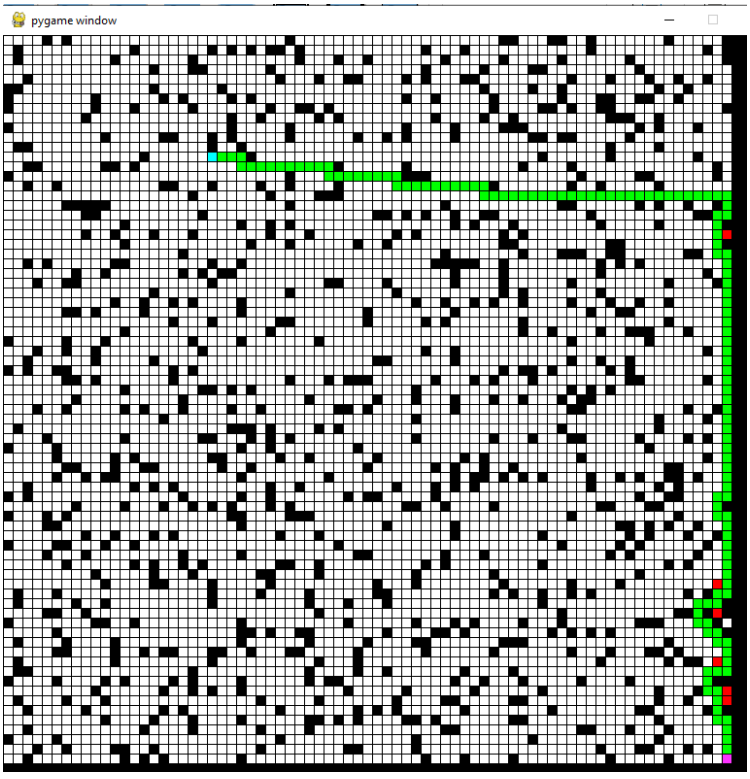


Figura 4 - Interface Gráfica *Greedy Search* *- Produto para Output

4.5.3. Interface Gráfica *BFS*

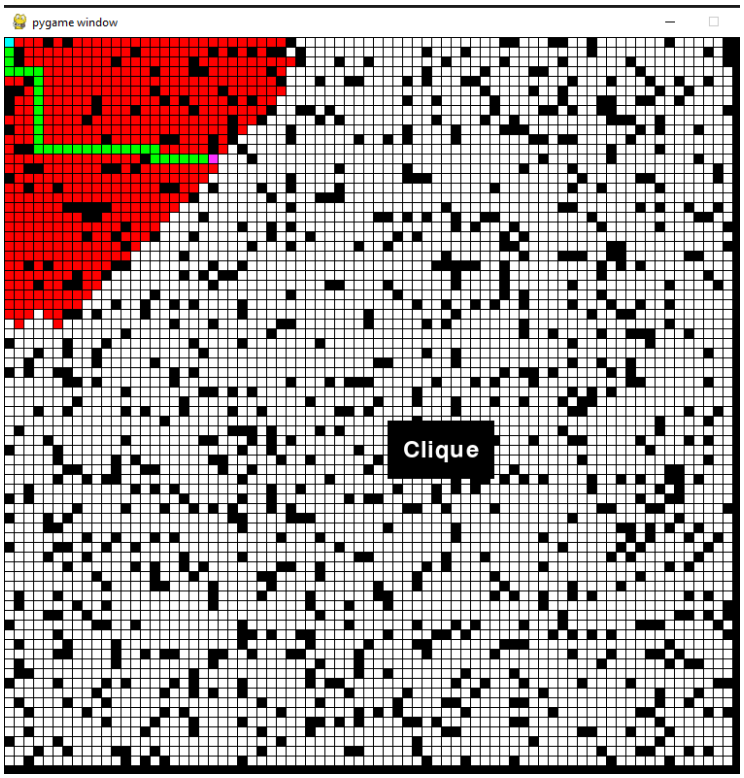


Figura 5 - Interface Gráfica *BFS* - Robô para Produto

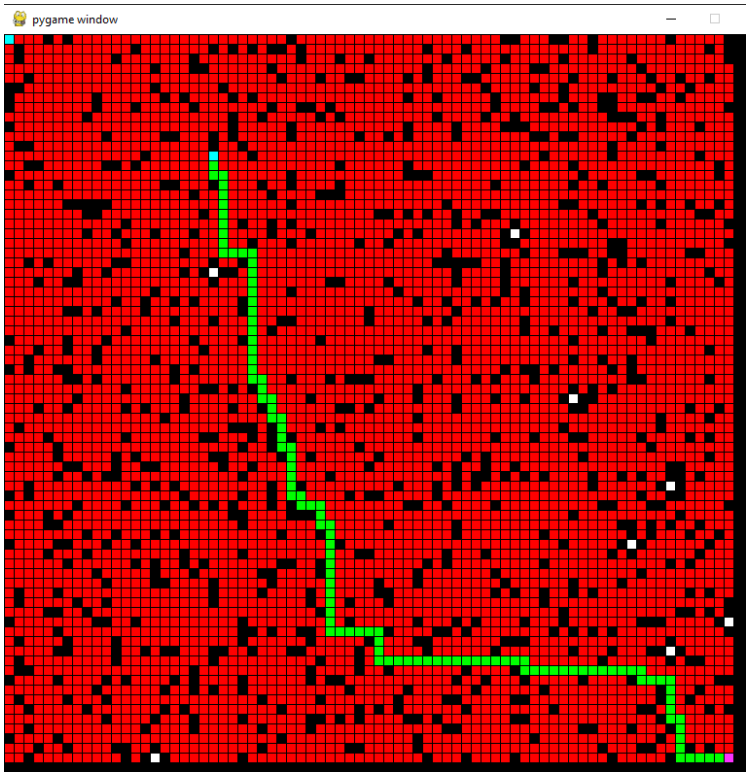


Figura 6 - Interface Gráfica *BFS* - Produto para Output

5. Resultados

Foram criadas 72 matrizes para simular os armazéns e avaliar o desempenho de cada algoritmo. As matrizes criadas são do tipo:

Número de Matrizes	Tamanho da Matriz	Tipo de Matriz
5	10x10	Nível Baixo de Obstáculos
	25x25	Nível Baixo de Obstáculos
	48x52	Nível Baixo de Obstáculos
	50x50	Nível Baixo de Obstáculos
	75x75	Nível Baixo de Obstáculos
	10x10	Nível Médio de Obstáculos
	25x25	Nível Médio de Obstáculos
	48x52	Nível Médio de Obstáculos
	50x50	Nível Médio de Obstáculos
	75x75	Nível Médio de Obstáculos
	10x10	Nível Alto de Obstáculos
	25x25	Nível Alto de Obstáculos
	48x52	Nível Alto de Obstáculos
	50x50	Nível Alto de Obstáculos
	75x75	Nível Alto de Obstáculos

Tabela 1 - Armazéns Simulados

Foi criado um procedimento para validar o desempenho de cada algoritmo em cada um destes armazéns. Para validar estes procedimentos foram retiradas várias métricas como, media, mediana, desvio padrão, máximo e mínimo, para verificar qual a performance de todos os algoritmos.

Os dados foram separados da seguinte forma:

- Custo de Robô para Produto (onde é calculado em unidades de tempo);
- Custo de Produto para Output (onde é calculado em unidades de tempo);
- Custo de Robô para Produto em segundos (estimando que cada casa o robô demora 0.1segundos a percorrer);

- Custo de Produto para Output em segundos (estimando que cada casa o robô demora 0.1segundos a percorrer);
- Tempo de procura do caminho do Robô para Produto em segundos;
- Tempo de procura do caminho do Produto para Output em segundos;

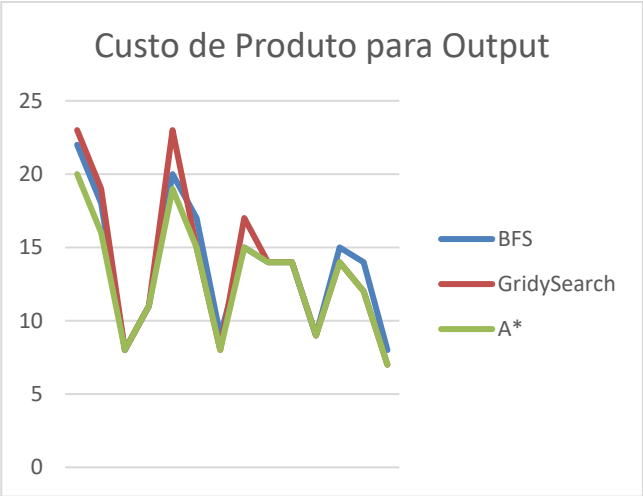
5.1. Resultados Matrizes 10x10

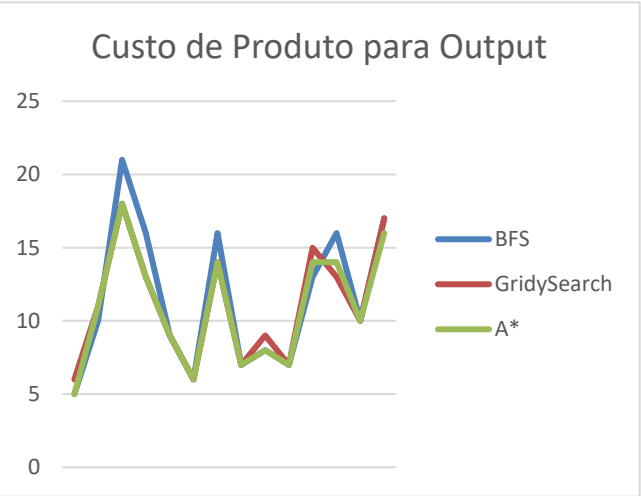
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_10x10_HighDensity_1.txt	BFS	22	5	2,2	0,5	0	0
MatrizRandom_10x10_HighDensity_2.txt	BFS	18	10	1,8	1	0	0
MatrizRandom_10x10_HighDensity_3.txt	BFS	8	21	0,8	2,1	0	0,00152564
MatrizRandom_10x10_HighDensity_4.txt	BFS	11	16	1,1	1,6	0,000890017	0,000999212
MatrizRandom_10x10_HighDensity_5.txt	BFS	20	9	2	0,9	0,000917912	0,001362562
MatrizRandom_10x10_Density_1.txt	BFS	17	6	1,7	0,6	0,000999928	0,000686169
MatrizRandom_10x10_Density_2.txt	BFS	9	16	0,9	1,6	0,000785589	0,000995636
MatrizRandom_10x10_Density_3.txt	BFS	15	7	1,5	0,7	0,001701117	0,000997782
MatrizRandom_10x10_Density_4.txt	BFS	14	8	1,4	0,8	0,001008511	0,000789881
MatrizRandom_10x10_Density_5.txt	BFS	14	7	1,4	0,7	0,001003981	0,000992537
MatrizRandom_10x10_MediumDensity_1.txt	BFS	9	13	0,9	1,3	0,001003742	0,001000404
MatrizRandom_10x10_MediumDensity_2.txt	BFS	15	16	1,5	1,6	0,000996351	0,001528502
MatrizRandom_10x10_MediumDensity_4.txt	BFS	14	10	1,4	1	0,001003265	0,001003981
MatrizRandom_10x10_MediumDensity_5.txt	BFS	8	17	0,8	1,7	0,001004696	0,000996828
Metricas	**Media**	13,85714286	11,5	1,385714286	1,15	0,000808222	0,000919938
	Mediana	14	10	1,4	1	0,000998139	0,000997305
	DesvioPadrão	4,290473548	4,777177888	0,429047355	0,477717789	0,00046611	0,000442402
	Maximo	22	21	2,2	2,1	0,001701117	0,001528502
	Minimo	8	5	0,8	0,5	0	0

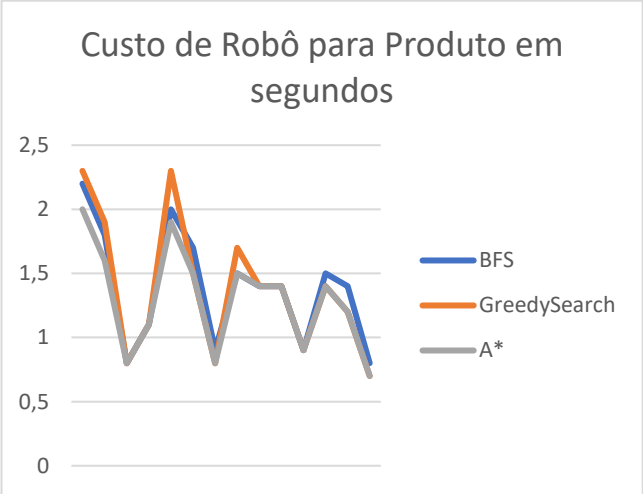
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_10x10_HighDensity_1.txt	GreedySearch	23	6	2,3	0,6	0	0
MatrizRandom_10x10_HighDensity_2.txt	GreedySearch	19	11	1,9	1,1	0,008353472	0,00099802
MatrizRandom_10x10_HighDensity_3.txt	GreedySearch	8	18	0,8	1,8	0,000999689	0,001018763
MatrizRandom_10x10_HighDensity_4.txt	GreedySearch	11	13	1,1	1,3	0,001312733	0
MatrizRandom_10x10_HighDensity_5.txt	GreedySearch	23	9	2,3	0,9	0,003252983	0,001006365
MatrizRandom_10x10_Density_1.txt	GreedySearch	15	6	1,5	0,6	0,001010418	0,000983953
MatrizRandom_10x10_Density_2.txt	GreedySearch	8	14	0,8	1,4	0	0
MatrizRandom_10x10_Density_3.txt	GreedySearch	17	7	1,7	0,7	0,000986099	0
MatrizRandom_10x10_Density_4.txt	GreedySearch	14	9	1,4	0,9	0,001582623	0
MatrizRandom_10x10_Density_5.txt	GreedySearch	14	7	1,4	0,7	0	0
MatrizRandom_10x10_MediumDensity_1.txt	GreedySearch	9	15	0,9	1,5	0	0
MatrizRandom_10x10_MediumDensity_2.txt	GreedySearch	14	13	1,4	1,3	0	0,000994444
MatrizRandom_10x10_MediumDensity_4.txt	GreedySearch	12	10	1,2	1	0,001002312	0,00100565
MatrizRandom_10x10_MediumDensity_5.txt	GreedySearch	7	17	0,7	1,7	0	0,000999451
Metricas	**Media**	13,85714286	11,07142857	1,385714286	1,107142857	0,001321452	0,000500475
	Mediana	14	10,5	1,4	1,05	0,000992894	0,000491977
	DesvioPadrão	5,040651076	3,863090652	0,504065108	0,386309065	0,002140434	0,000500525
	Maximo	23	18	2,3	1,8	0,008353472	0,001018763
	Minimo	7	6	0,7	0,6	0	0

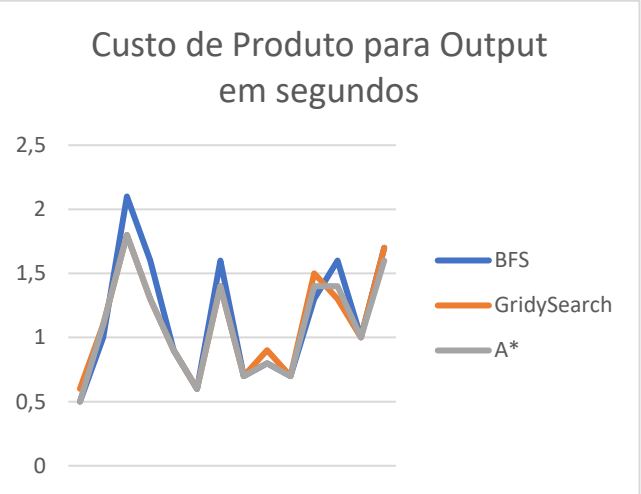
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_10x10_HighDensity_1.txt	A*	20	5	2	0,5	0,000998735	0,00099802
MatrizRandom_10x10_HighDensity_2.txt	A*	16	11	1,6	1,1	0	0,001734018
MatrizRandom_10x10_HighDensity_3.txt	A*	8	18	0,8	1,8	0,000998735	0,001010418
MatrizRandom_10x10_HighDensity_4.txt	A*	11	13	1,1	1,3	0,000991821	0,000999928
MatrizRandom_10x10_HighDensity_5.txt	A*	19	9	1,9	0,9	0,001000404	0,000999689
MatrizRandom_10x10_Density_1.txt	A*	15	6	1,5	0,6	0,000999689	0,001002312
MatrizRandom_10x10_Density_2.txt	A*	8	14	0,8	1,4	0	0,00100112
MatrizRandom_10x10_Density_3.txt	A*	15	7	1,5	0,7	0,00199914	0,000999928
MatrizRandom_10x10_Density_4.txt	A*	14	8	1,4	0,8	0,001543045	0,000484228
MatrizRandom_10x10_Density_5.txt	A*	14	7	1,4	0,7	0,002772331	0,000997066
MatrizRandom_10x10_MediumDensity_1.txt	A*	9	14	0,9	1,4	0,001000404	0,001998425
MatrizRandom_10x10_MediumDensity_2.txt	A*	14	14	1,4	1,4	0,001000404	0,001000166
MatrizRandom_10x10_MediumDensity_4.txt	A*	12	10	1,2	1	0,001001835	0,001000881
MatrizRandom_10x10_MediumDensity_5.txt	A*	7	16	0,7	1,6	0,000629187	0
Metricas	**Media**	13	10,85714286	1,3	1,085714286	0,001066838	0,001016157
	Mediana	14	10,5	1,4	1,05	0,001000047	0,001000047
	DesvioPadrão	3,891382421	3,888759311	0,389138242	0,388875931	0,000681876	0,001000524
	Maximo	20	18	2	1,8	0,002772331	0,001000106
	Minimo	7	5	0,7	0,5	0	0,001000106

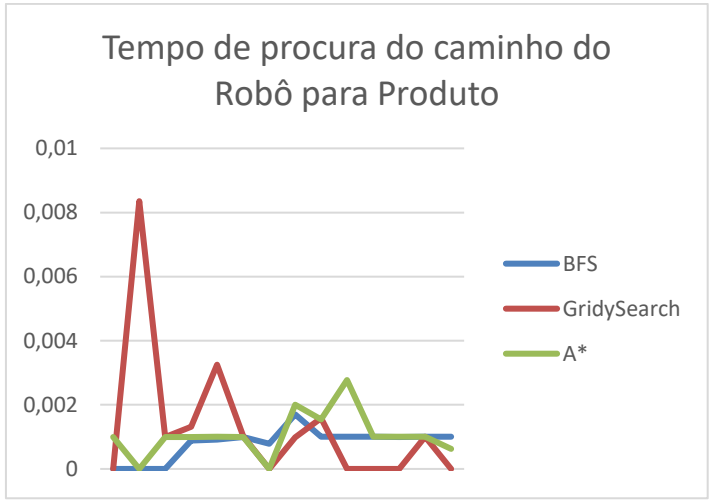
Figura 7 - Resultados Matrizes 10x10

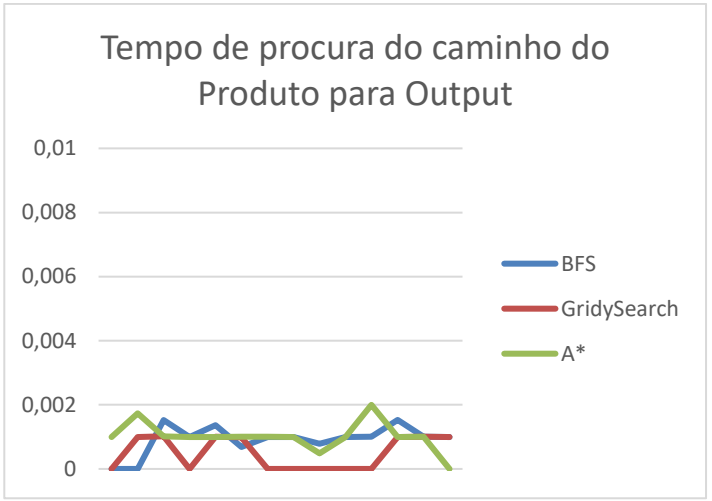

Custo de Produto para Output
25
20
15
10
5
0
BFS
GridySearch
A*


Custo de Produto para Output
25
20
15
10
5
0
BFS
GridySearch
A*


Custo de Robô para Produto em segundos
2,5
2
1,5
1
0,5
0
BFS
GreedySearch
A*


Custo de Produto para Output em segundos
2,5
2
1,5
1
0,5
0
BFS
GridySearch
A*





	BFS	*Greedy Search*	A*
Tempo Total Médio (Tempo Movimento do Robô + Tempo de Processamento)	2,537442446	2,49467907	2,387797281

Tabela 2 – Tempo Total Médio de cada Algoritmo em Matrizes 10x10

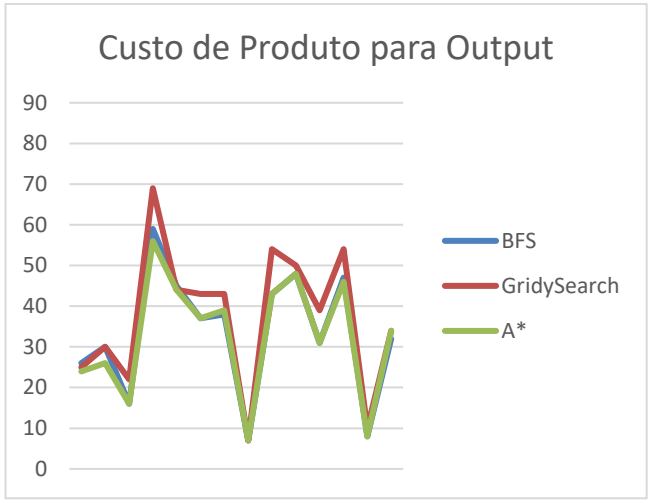
5.2. Resultados Matrizes 25x25

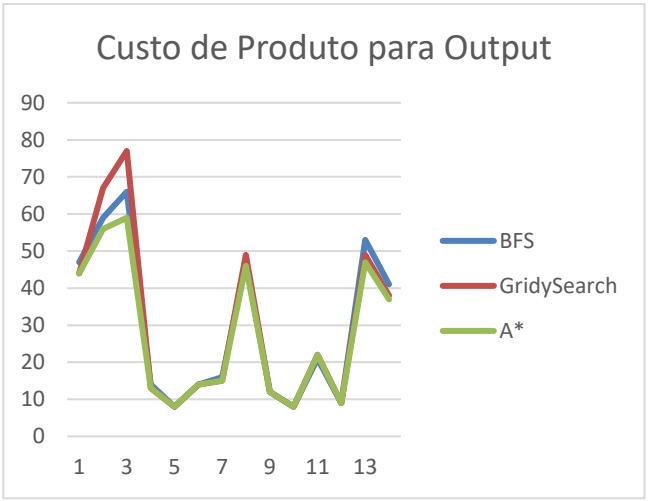
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_25x25_HighDensity_1.txt	BFS	26	47	2,6	4,7	0,001003265	0,001985073
MatrizRandom_25x25_HighDensity_2.txt	BFS	30	59	3	5,9	0,00200057	0,002097845
MatrizRandom_25x25_HighDensity_3.txt	BFS	16	66	1,6	6,6	0,00099802	0,002033949
MatrizRandom_25x25_HighDensity_5.txt	BFS	59	14	5,9	1,4	0,002820253	0,001045704
MatrizRandom_25x25_LowDensity_1.txt	BFS	45	8	4,5	0,8	0	0
MatrizRandom_25x25_LowDensity_2.txt	BFS	37	14	3,7	1,4	0,004118443	0,001881838
MatrizRandom_25x25_LowDensity_3.txt	BFS	38	16	3,8	1,6	0,001997948	0,001411915
MatrizRandom_25x25_LowDensity_4.txt	BFS	7	47	0,7	4,7	0,001389742	0,00199914
MatrizRandom_25x25_LowDensity_5.txt	BFS	43	12	4,3	1,2	0,00207448	0,000782728
MatrizRandom_25x25_MediumDensity_1.txt	BFS	48	8	4,8	0,8	0,001996279	0,0006423
MatrizRandom_25x25_MediumDensity_2.txt	BFS	31	21	3,1	2,1	0,00134635	0,000946045
MatrizRandom_25x25_MediumDensity_3.txt	BFS	47	9	4,7	0,9	0,003709078	0,00054884
MatrizRandom_25x25_MediumDensity_4.txt	BFS	8	53	0,8	5,3	0	0,001582623
MatrizRandom_25x25_MediumDensity_5.txt	BFS	32	41	3,2	4,1	0	0
Metricas	Media	33,35714286	29,64285714	3,335714286	2,964285714	0,001675316	0,001211286
	Mediana	34,5	18,5	3,45	1,85	0,00169301	0,001228809
	DesvioPadrão	14,71931259	20,50298634	1,471931259	2,050298634	0,001236596	0,000721528
	Maximo	59	66	5,9	6,6	0,004118443	0,002097845
	Minimo	7	8	0,7	0,8	0	0

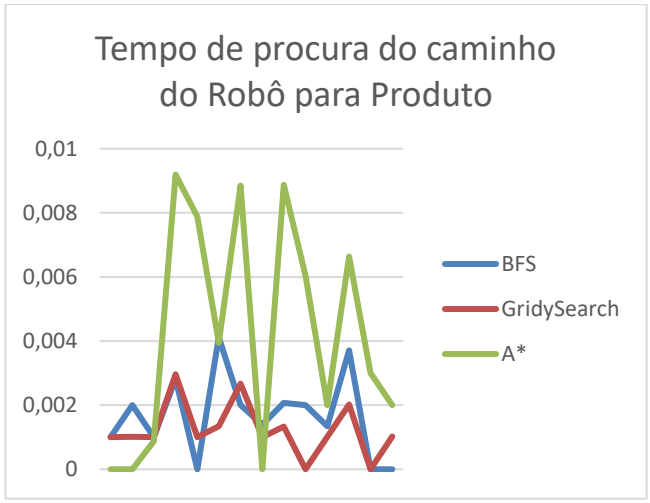
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_25x25_HighDensity_1.txt	GreedySearch	25	44	2,5	4,4	0,001003504	0,000995159
MatrizRandom_25x25_HighDensity_2.txt	GreedySearch	30	67	3	6,7	0,001012325	0,001001835
MatrizRandom_25x25_HighDensity_3.txt	GreedySearch	22	77	2,2	7,7	0,001003027	0,002007723
MatrizRandom_25x25_HighDensity_5.txt	GreedySearch	69	13	6,9	1,3	0,00296402	0,001507998
MatrizRandom_25x25_LowDensity_1.txt	GreedySearch	44	8	4,4	0,8	0,001003981	0
MatrizRandom_25x25_LowDensity_2.txt	GreedySearch	43	14	4,3	1,4	0,001337767	0,000192881
MatrizRandom_25x25_LowDensity_3.txt	GreedySearch	43	15	4,3	1,5	0,002666712	0
MatrizRandom_25x25_LowDensity_4.txt	GreedySearch	7	49	0,7	4,9	0,001000404	0,001028776
MatrizRandom_25x25_LowDensity_5.txt	GreedySearch	54	12	5,4	1,2	0,001333475	0
MatrizRandom_25x25_MediumDensity_1.txt	GreedySearch	50	8	5	0,8	0	0,004128933
MatrizRandom_25x25_MediumDensity_2.txt	GreedySearch	39	22	3,9	2,2	0,00101018	0
MatrizRandom_25x25_MediumDensity_3.txt	GreedySearch	54	9	5,4	0,9	0,002024651	0
MatrizRandom_25x25_MediumDensity_4.txt	GreedySearch	10	49	1	4,9	0	0
MatrizRandom_25x25_MediumDensity_5.txt	GreedySearch	34	38	3,4	3,8	0,001025677	0,000974417
Metricas	Media	37,42857143	30,35714286	3,742857143	3,035714286	0,001241837	0,000845551
	Mediana	41	18,5	4,1	1,85	0,001011252	0,000583649
	DesvioPadrão	16,79589357	22,55978452	1,679589357	2,255978452	0,000805901	0,001114283
	Maximo	69	77	6,9	7,7	0,00296402	0,004128933
	Minimo	7	8	0,7	0,8	0	0

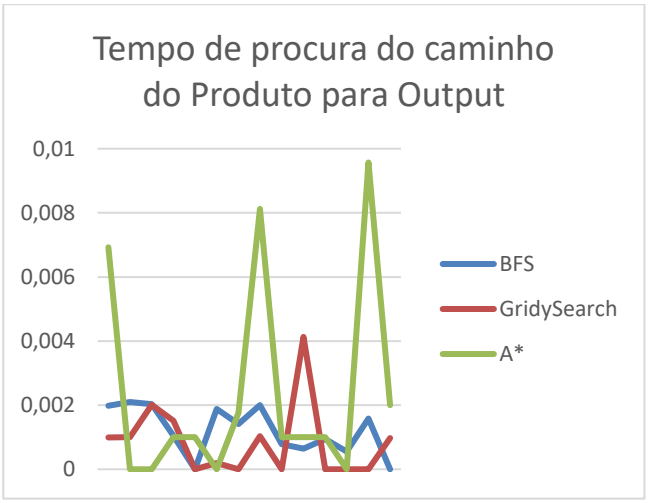
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_25x25_HighDensity_1.txt	A*	24	44	2,4	4,4	0	0,00692296
MatrizRandom_25x25_HighDensity_2.txt	A*	26	56	2,6	5,6	0	0
MatrizRandom_25x25_HighDensity_3.txt	A*	16	59	1,6	5,9	0,000874043	0
MatrizRandom_25x25_HighDensity_5.txt	A*	56	13	5,6	1,3	0,009191513	0,00100112
MatrizRandom_25x25_LowDensity_1.txt	A*	44	8	4,4	0,8	0,007882357	0,00099802
MatrizRandom_25x25_LowDensity_2.txt	A*	37	14	3,7	1,4	0,00395155	0
MatrizRandom_25x25_LowDensity_3.txt	A*	39	15	3,9	1,5	0,008852959	0,001758337
MatrizRandom_25x25_LowDensity_4.txt	A*	7	46	0,7	4,6	0	0,008119822
MatrizRandom_25x25_LowDensity_5.txt	A*	43	12	4,3	1,2	0,008868933	0,000999689
MatrizRandom_25x25_MediumDensity_1.txt	A*	48	8	4,8	0,8	0,00603199	0,000998974
MatrizRandom_25x25_MediumDensity_2.txt	A*	31	22	3,1	2,2	0,002000093	0,001001835
MatrizRandom_25x25_MediumDensity_3.txt	A*	46	9	4,6	0,9	0,006631136	0
MatrizRandom_25x25_MediumDensity_4.txt	A*	8	47	0,8	4,7	0,003000736	0,009569883
MatrizRandom_25x25_MediumDensity_5.txt	A*	34	37	3,4	3,7	0,00200057	0,00200057
Metricas	Media	32,78571429	27,85714286	3,278571429	2,785714286	0,004234706	0,002383658
	Mediana	35,5	18,5	3,55	1,85	0,003476143	0,001000404
	DesvioPadrão	14,4428854	18,53512936	1,44428854	1,853512936	0,003446869	0,003139487
	Maximo	56	59	5,6	5,9	0,009191513	0,009569883
	Minimo	7	8	0,7	0,8	0	0

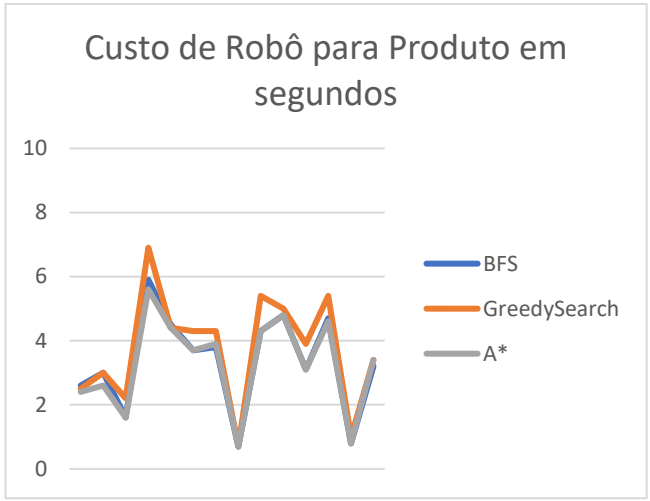
Figura 8 - Resultados Matrizes 25x25

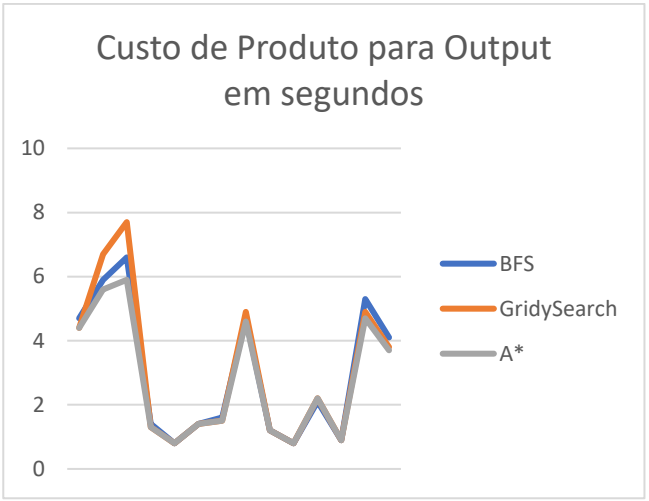

Custo de Produto para Output
90
80
70
60
50
40
30
20
10
0
BFS
GridySearch
A*


Custo de Produto para Output
90
80
70
60
50
40
30
20
10
0
1
3
5
7
9
11
13
BFS
GridySearch
A*


Tempo de procura do caminho do Robô para Produto
0,01
0,008
0,006
0,004
0,002
0
BFS
GridySearch
A*


Tempo de procura do caminho do Produto para Output
0,01
0,008
0,006
0,004
0,002
0
BFS
GridySearch
A*





	BFS	*Greedy Search*	A*
Tempo Total Médio (Tempo Movimento do Robô + Tempo de Processamento)	6,302886602	6,780658817	6,070904078

Tabela 3 - Tempo Total Médio de cada Algoritmo em Matrizes 25x25

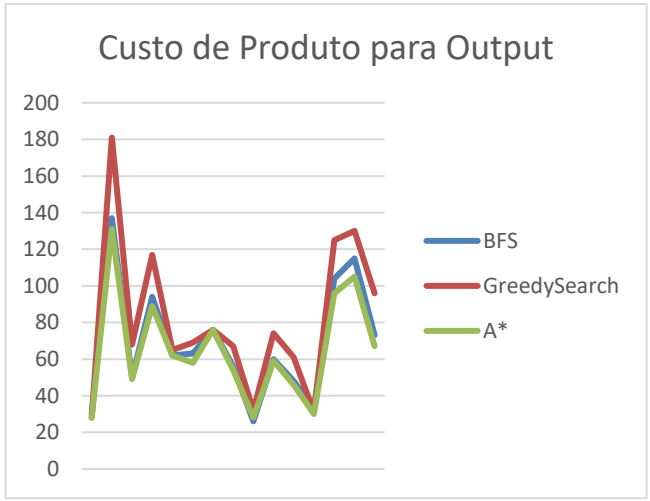
5.3. Resultados Matrizes 48x52

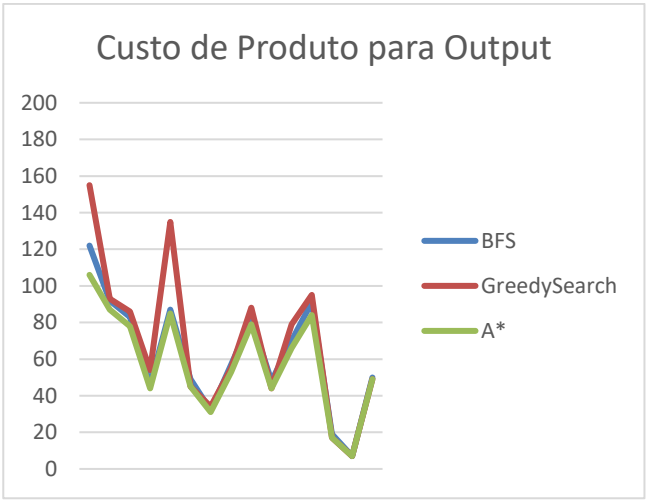
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_48x52_HighDensity_1.txt	BFS	30	122	3	12,2	0,006169081	0,004999399
MatrizRandom_48x52_HighDensity_2.txt	BFS	137	92	13,7	9,2	0,005996943	0,003019094
MatrizRandom_48x52_HighDensity_3.txt	BFS	50	83	5	8,3	0,001973391	0,003645182
MatrizRandom_48x52_HighDensity_4.txt	BFS	94	46	9,4	4,6	0,005258799	0,003000736
MatrizRandom_48x52_HighDensity_5.txt	BFS	62	87	6,2	8,7	0,00125885	0,003996134
MatrizRandom_48x52_LowDensity_1.txt	BFS	63	49	6,3	4,9	0,004608154	0,006639004
MatrizRandom_48x52_LowDensity_2.txt	BFS	76	32	7,6	3,2	0,005965233	0,002988338
MatrizRandom_48x52_LowDensity_3.txt	BFS	56	57	5,6	5,7	0,003830194	0,004523516
MatrizRandom_48x52_LowDensity_4.txt	BFS	26	82	2,6	8,2	0,001286268	0,009223938
MatrizRandom_48x52_LowDensity_5.txt	BFS	60	48	6	4,8	0,008017302	0,004263639
MatrizRandom_48x52_MediumDensity_1.txt	BFS	48	70	4,8	7	0,005151987	0,007902861
MatrizRandom_48x52_MediumDensity_2.txt	BFS	33	91	3,3	9,1	0,001006126	0,004792452
MatrizRandom_48x52_MediumDensity_3.txt	BFS	104	19	10,4	1,9	0	0,006459713
MatrizRandom_48x52_MediumDensity_4.txt	BFS	115	7	11,5	0,7	0,006992817	0,001001596
MatrizRandom_48x52_MediumDensity_5.txt	BFS	73	50	7,3	5	0,002710342	0,006000042
Metricas	Media	68,46666667	62,33333333	6,846666667	6,233333333	0,004015032	0,004830376
	Mediana	62	57	6,2	5,7	0,004608154	0,004523516
	DesvioPadrão	31,04162081	29,92583425	3,104162081	2,992583425	0,002402351	0,002053688
	Maximo	137	122	13,7	12,2	0,008017302	0,009223938
	Minimo	26	7	2,6	0,7	0	0,001001596

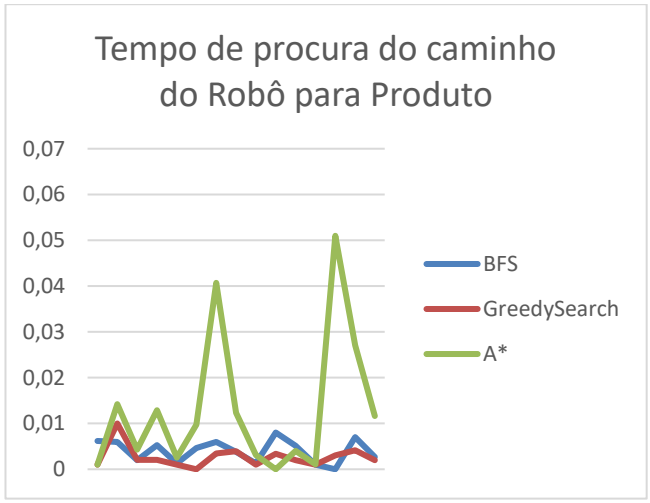
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_48x52_HighDensity_1.txt	GreedySearch	28	155	2,8	15,5	0,001028061	0,000412464
MatrizRandom_48x52_HighDensity_2.txt	GreedySearch	181	93	18,1	9,3	0,009998798	0,002557278
MatrizRandom_48x52_HighDensity_3.txt	GreedySearch	68	86	6,8	8,6	0,002031565	0,001024723
MatrizRandom_48x52_HighDensity_4.txt	GreedySearch	117	54	11,7	5,4	0,002066135	0
MatrizRandom_48x52_HighDensity_5.txt	GreedySearch	65	135	6,5	13,5	0,001027822	0,002999306
MatrizRandom_48x52_LowDensity_1.txt	GreedySearch	69	45	6,9	4,5	0	0
MatrizRandom_48x52_LowDensity_2.txt	GreedySearch	76	34	7,6	3,4	0,003479004	0,000879526
MatrizRandom_48x52_LowDensity_3.txt	GreedySearch	67	55	6,7	5,5	0,003932953	0,00163722
MatrizRandom_48x52_LowDensity_4.txt	GreedySearch	32	88	3,2	8,8	0,000978708	0,002981424
MatrizRandom_48x52_LowDensity_5.txt	GreedySearch	74	44	7,4	4,4	0,003368855	0,002051592
MatrizRandom_48x52_MediumDensity_1.txt	GreedySearch	61	79	6,1	7,9	0,002001047	0,002024889
MatrizRandom_48x52_MediumDensity_2.txt	GreedySearch	31	95	3,1	9,5	0,000999928	0,001998186
MatrizRandom_48x52_MediumDensity_3.txt	GreedySearch	125	17	12,5	1,7	0,00306201	0,000854969
MatrizRandom_48x52_MediumDensity_4.txt	GreedySearch	130	7	13	0,7	0,004142761	0
MatrizRandom_48x52_MediumDensity_5.txt	GreedySearch	96	49	9,6	4,9	0,001966476	0,000262022
Metricas	Media	81,33333333	69,06666667	8,133333333	6,906666667	0,002672275	0,00131224
	Mediana	69	55	6,9	5,5	0,002031565	0,001024723
	DesvioPadrão	40,66557376	39,62232816	4,066557376	3,962232816	0,002289946	0,001044688
	Maximo	181	155	18,1	15,5	0,009998798	0,002999306
	Minimo	28	7	2,8	0,7	0	0

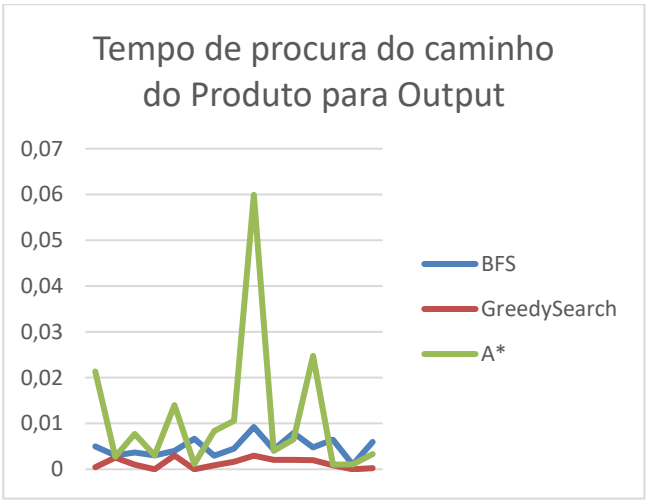
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_48x52_HighDensity_1.txt	A*	28	106	2,8	10,6	0,000975847	0,021332979
MatrizRandom_48x52_HighDensity_2.txt	A*	131	87	13,1	8,7	0,014199734	0,002766609
MatrizRandom_48x52_HighDensity_3.txt	A*	49	78	4,9	7,8	0,004319906	0,007717848
MatrizRandom_48x52_HighDensity_4.txt	A*	89	44	8,9	4,4	0,012856483	0,002998829
MatrizRandom_48x52_HighDensity_5.txt	A*	62	85	6,2	8,5	0,002530098	0,01403141
MatrizRandom_48x52_LowDensity_1.txt	A*	58	45	5,8	4,5	0,009738445	0,001123905
MatrizRandom_48x52_LowDensity_2.txt	A*	76	31	7,6	3,1	0,040691853	0,008335352
MatrizRandom_48x52_LowDensity_3.txt	A*	54	53	5,4	5,3	0,012303352	0,01060915
MatrizRandom_48x52_LowDensity_4.txt	A*	28	79	2,8	7,9	0,003134251	0,059929371
MatrizRandom_48x52_LowDensity_5.txt	A*	59	44	5,9	4,4	0	0,004036427
MatrizRandom_48x52_MediumDensity_1.txt	A*	46	66	4,6	6,6	0,004017115	0,006520271
MatrizRandom_48x52_MediumDensity_2.txt	A*	30	84	3	8,4	0,001047373	0,024760485
MatrizRandom_48x52_MediumDensity_3.txt	A*	96	17	9,6	1,7	0,050984621	0,001010418
MatrizRandom_48x52_MediumDensity_4.txt	A*	105	7	10,5	0,7	0,027059793	0,001052618
MatrizRandom_48x52_MediumDensity_5.txt	A*	67	49	6,7	4,9	0,011653423	0,003336668
Metricas	Media	65,2	58,33333333	6,52	5,833333333	0,013034153	0,011304156
	Mediana	59	53	5,9	5,3	0,009738445	0,006520271
	DesvioPadrão	28,76618385	27,2413085	2,876618385	2,72413085	0,01469957	0,014767995
	Maximo	131	106	13,1	10,6	0,050984621	0,059929371
	Minimo	28	7	2,8	0,7	0	0,001010418

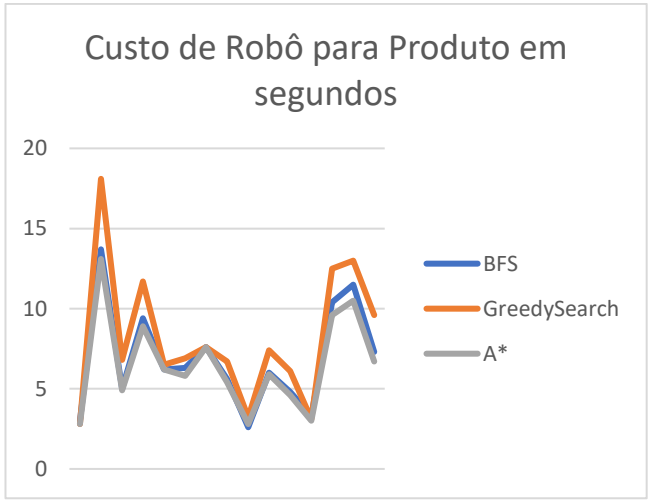
Figura 9 - Resultados Matrizes 48x52

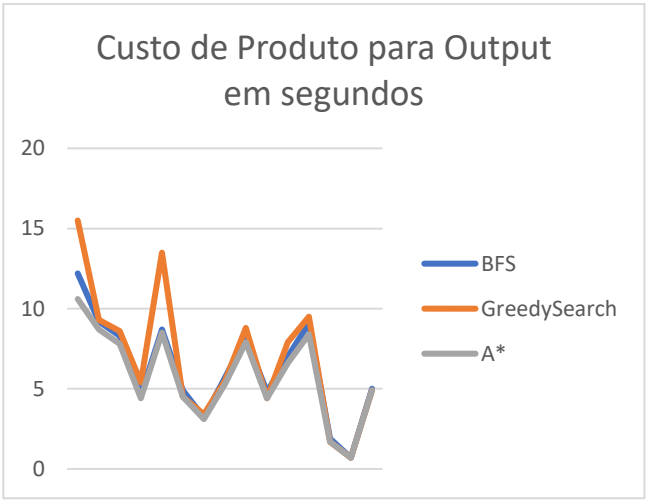

Custo de Produto para Output
200
180
160
140
120
100
80
60
40
20
0
BFS
GreedySearch
A*


Custo de Produto para Output
200
180
160
140
120
100
80
60
40
20
0
BFS
GreedySearch
A*


Tempo de procura do caminho do Robô para Produto
0,07
0,06
0,05
0,04
0,03
0,02
0,01
0
BFS
GreedySearch
A*


Tempo de procura do caminho do Produto para Output
0,07
0,06
0,05
0,04
0,03
0,02
0,01
0
BFS
GreedySearch
A*





	BFS	*Greedy Search*	A*
Tempo Total Médio (Tempo Movimento do Robô + Tempo de Processamento)	13,08884541	15,04398451	12,37767164

Tabela 4 - Tempo Total Médio de cada Algoritmo em Matrizes 48x52

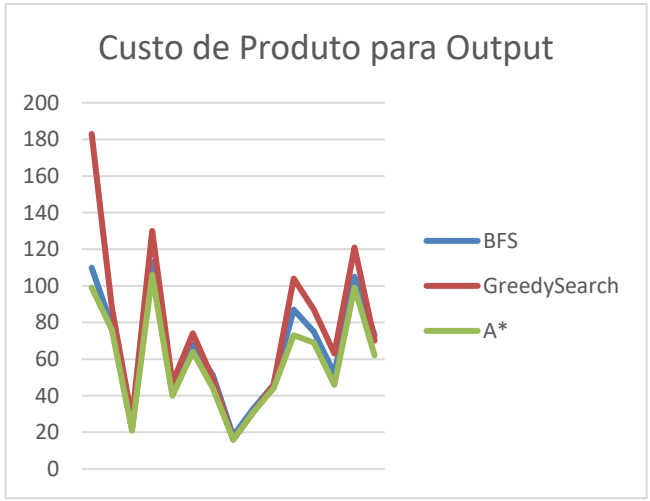
5.4. Resultados Matrizes 50x50

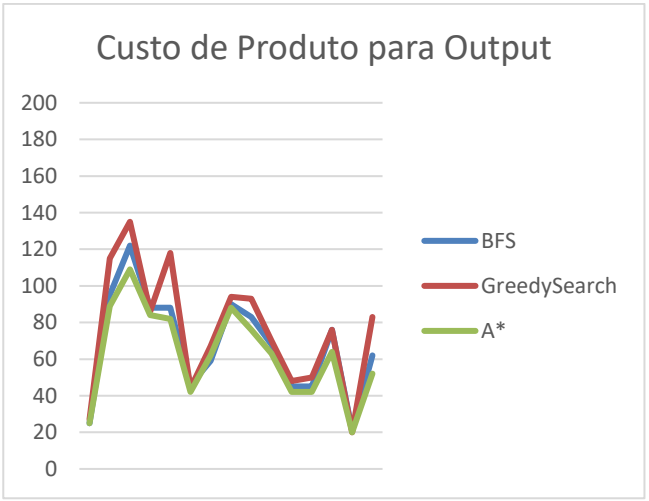
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_50x50_HighDensity_1.txt	BFS	110	25	11	2,5	0,003806591	0,001001596
MatrizRandom_50x50_HighDensity_2.txt	BFS	78	95	7,8	9,5	0,004942179	0,003881693
MatrizRandom_50x50_HighDensity_3.txt	BFS	22	122	2,2	12,2	0,000996113	0,004391909
MatrizRandom_50x50_HighDensity_4.txt	BFS	113	88	11,3	8,8	0,006362677	0,002998829
MatrizRandom_50x50_HighDensity_5.txt	BFS	42	88	4,2	8,8	0,001601696	0,008112669
MatrizRandom_50x50_LowDensity_1.txt	BFS	68	46	6,8	4,6	0,003999472	0,001440525
MatrizRandom_50x50_LowDensity_2.txt	BFS	51	59	5,1	5,9	0,001026154	0,008051634
MatrizRandom_50x50_LowDensity_3.txt	BFS	18	90	1,8	9	0,000998974	0,006449699
MatrizRandom_50x50_LowDensity_4.txt	BFS	33	83	3,3	8,3	0,002090931	0,00600028
MatrizRandom_50x50_LowDensity_5.txt	BFS	46	68	4,6	6,8	0	0,002017021
MatrizRandom_50x50_MediumDensity_1.txt	BFS	87	45	8,7	4,5	0,008047819	0,004663706
MatrizRandom_50x50_MediumDensity_2.txt	BFS	75	45	7,5	4,5	0,004005671	0,0042274
MatrizRandom_50x50_MediumDensity_3.txt	BFS	52	76	5,2	7,6	0	0,013703823
MatrizRandom_50x50_MediumDensity_4.txt	BFS	105	21	10,5	2,1	0,010515213	0,000984669
MatrizRandom_50x50_MediumDensity_5.txt	BFS	73	62	7,3	6,2	0,002137899	0,002906322
Metricas	Media	64,86666667	67,53333333	6,486666667	6,753333333	0,003368759	0,004722118
	Mediana	68	68	6,8	6,8	0,002137899	0,0042274
	DesvioPadrão	29,54288784	26,99596678	2,954288784	2,699596678	0,002956353	0,003277177
	Maximo	113	122	11,3	12,2	0,010515213	0,013703823
	Minimo	18	21	1,8	2,1	0	0,000984669

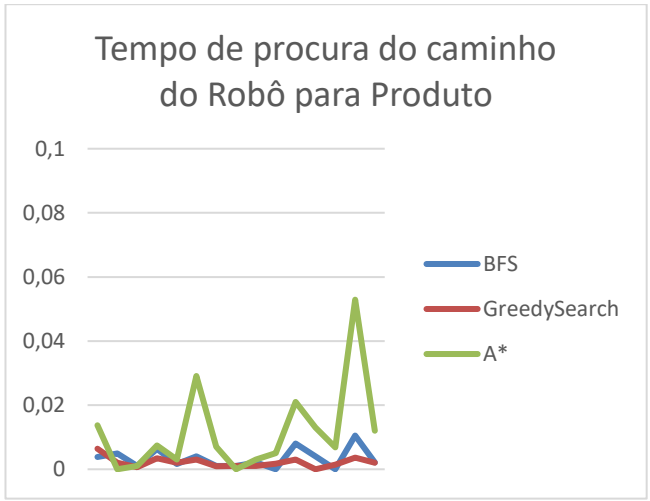
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_50x50_HighDensity_1.txt	GreedySearch	183	27	18,3	2,7	0,006413221	0,001359463
MatrizRandom_50x50_HighDensity_2.txt	GreedySearch	88	115	8,8	11,5	0,001998663	0,001233816
MatrizRandom_50x50_HighDensity_3.txt	GreedySearch	27	135	2,7	13,5	0,00061202	0,003218412
MatrizRandom_50x50_HighDensity_4.txt	GreedySearch	130	86	13	8,6	0,003413677	0
MatrizRandom_50x50_HighDensity_5.txt	GreedySearch	47	118	4,7	11,8	0,002007961	0,002007008
MatrizRandom_50x50_LowDensity_1.txt	GreedySearch	74	44	7,4	4,4	0,003004789	0,002146244
MatrizRandom_50x50_LowDensity_2.txt	GreedySearch	49	67	4,9	6,7	0,000927687	0,003551006
MatrizRandom_50x50_LowDensity_3.txt	GreedySearch	16	94	1,6	9,4	0,001002789	0,003997564
MatrizRandom_50x50_LowDensity_4.txt	GreedySearch	31	93	3,1	9,3	0,001000166	0,002043486
MatrizRandom_50x50_LowDensity_5.txt	GreedySearch	46	70	4,6	7	0,00178051	0
MatrizRandom_50x50_MediumDensity_1.txt	GreedySearch	104	48	10,4	4,8	0,00300312	0,001929522
MatrizRandom_50x50_MediumDensity_2.txt	GreedySearch	87	50	8,7	5	0	0
MatrizRandom_50x50_MediumDensity_3.txt	GreedySearch	63	76	6,3	7,6	0,001485825	0,002228022
MatrizRandom_50x50_MediumDensity_4.txt	GreedySearch	121	20	12,1	2	0,003588915	0,001220942
MatrizRandom_50x50_MediumDensity_5.txt	GreedySearch	70	83	7	8,3	0,002025604	0,002143383
Metricas	Media	75,73333333	75,06666667	7,573333333	7,506666667	0,002150997	0,001805258
	Mediana	70	76	7	7,6	0,001998663	0,002007008
	DesvioPadrão	43,31507308	32,30369776	4,331507308	3,230369776	0,001528773	0,001183685
	Maximo	183	135	18,3	13,5	0,006413221	0,003997564
	Minimo	16	20	1,6	2	0	0

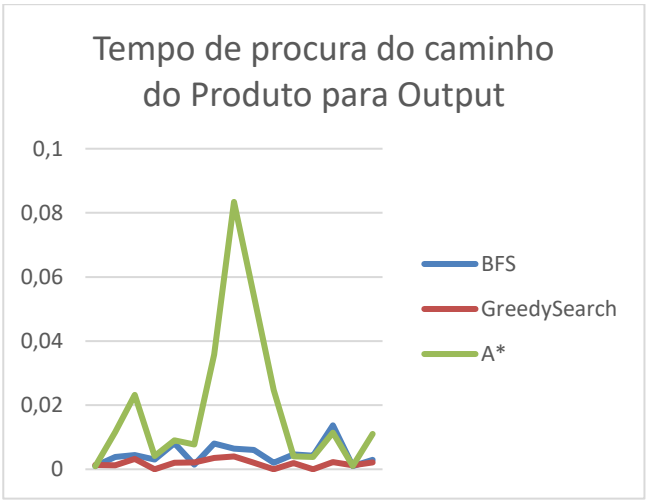
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_50x50_HighDensity_1.txt	A*	99	25	9,9	2,5	0,013679028	0,000947237
MatrizRandom_50x50_HighDensity_2.txt	A*	76	89	7,6	8,9	0	0,011641026
MatrizRandom_50x50_HighDensity_3.txt	A*	21	109	2,1	10,9	0,000999451	0,023168564
MatrizRandom_50x50_HighDensity_4.txt	A*	106	84	10,6	8,4	0,007411957	0,004085779
MatrizRandom_50x50_HighDensity_5.txt	A*	40	82	4	8,2	0,002999306	0,008999586
MatrizRandom_50x50_LowDensity_1.txt	A*	64	42	6,4	4,2	0,029103279	0,007718086
MatrizRandom_50x50_LowDensity_2.txt	A*	44	62	4,4	6,2	0,006997824	0,035771132
MatrizRandom_50x50_LowDensity_3.txt	A*	16	88	1,6	8,8	0	0,083414793
MatrizRandom_50x50_LowDensity_4.txt	A*	31	76	3,1	7,6	0,00299859	0,054158688
MatrizRandom_50x50_LowDensity_5.txt	A*	44	63	4,4	6,3	0,004999399	0,02466774
MatrizRandom_50x50_MediumDensity_1.txt	A*	73	42	7,3	4,2	0,020961285	0,004000664
MatrizRandom_50x50_MediumDensity_2.txt	A*	69	42	6,9	4,2	0,012998104	0,00387764
MatrizRandom_50x50_MediumDensity_3.txt	A*	46	64	4,6	6,4	0,006870985	0,011469603
MatrizRandom_50x50_MediumDensity_4.txt	A*	99	20	9,9	2	0,052909136	0,001001358
MatrizRandom_50x50_MediumDensity_5.txt	A*	62	52	6,2	5,2	0,011999369	0,011003017
Metricas	Media	59,33333333	62,66666667	5,933333333	6,266666667	0,011661847	0,019061661
	Mediana	62	63	6,2	6,3	0,006997824	0,011003017
	DesvioPadrão	27,18251072	24,6648648	2,718251072	2,46648648	0,013517696	0,022277296
	Maximo	106	109	10,6	10,9	0,052909136	0,083414793
	Minimo	16	20	1,6	2	0	0,000947237

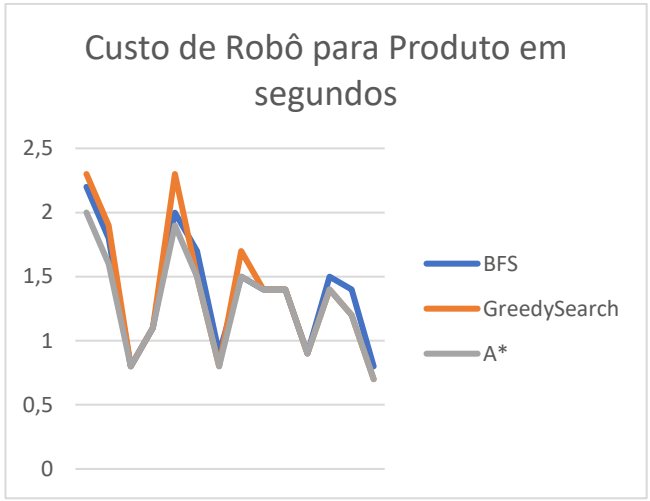
Figura 10 - Resultados Matrizes 50x50

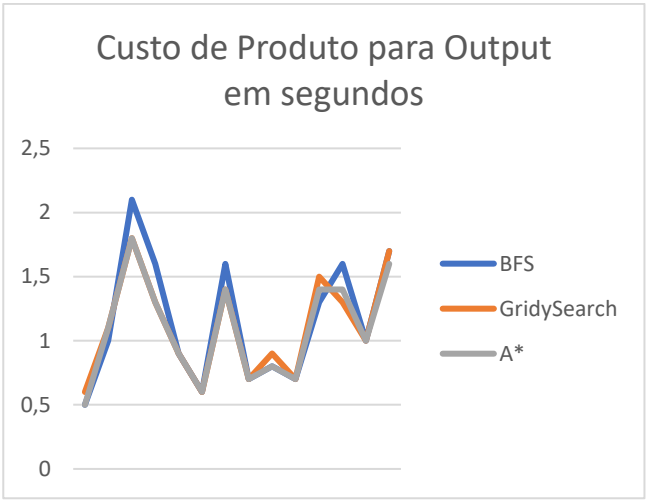

Custo de Produto para Output
200
180
160
140
120
100
80
60
40
20
0
BFS
GreedySearch
A*


Custo de Produto para Output
200
180
160
140
120
100
80
60
40
20
0
BFS
GreedySearch
A*


Tempo de procura do caminho do Robô para Produto
0,1
0,08
0,06
0,04
0,02
0
BFS
GreedySearch
A*


Tempo de procura do caminho do Produto para Output
0,1
0,08
0,06
0,04
0,02
0
BFS
GreedySearch
A*





	BFS	*Greedy Search*	A*
Tempo Total Médio (Tempo Movimento do Robô + Tempo de Processamento)	13,24809088	15,08395625	12,23072351

Tabela 5 - Tempo Total Médio de cada Algoritmo em Matrizes 50x50

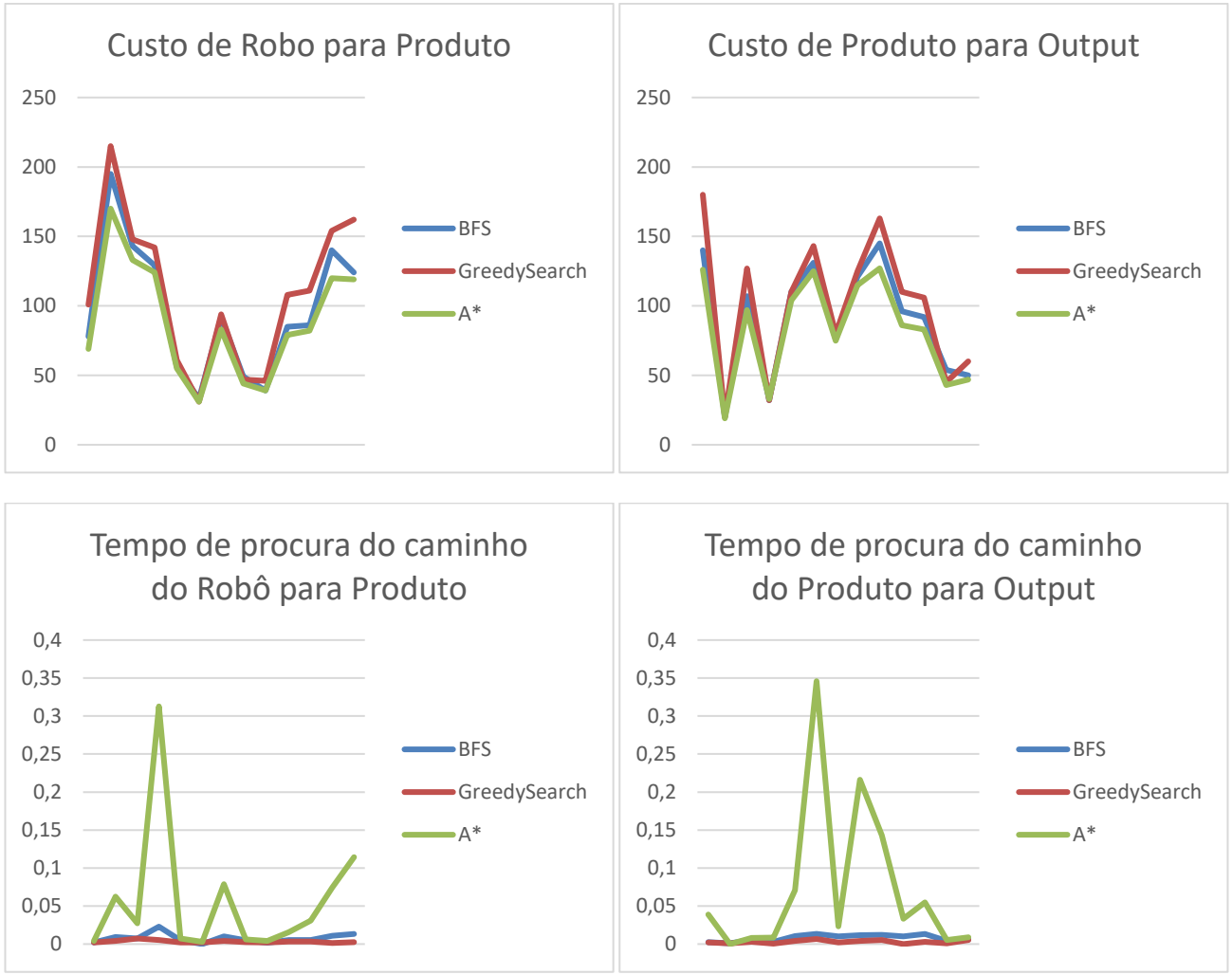
5.5. Resultados Matrizes 75x75

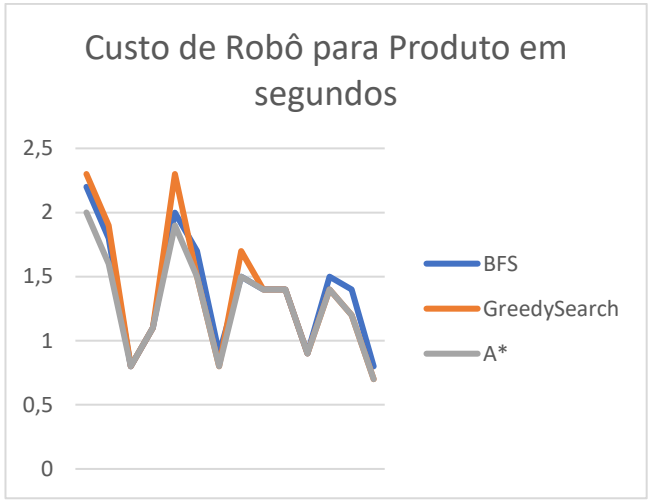
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_75x75_HighDensity_1.txt	BFS	78	140	7,8	14	0,001999617	0,00235486
MatrizRandom_75x75_HighDensity_2.txt	BFS	195	20	19,5	2	0,009520292	0,000999689
MatrizRandom_75x75_HighDensity_3.txt	BFS	143	107	14,3	10,7	0,007504225	0,00559783
MatrizRandom_75x75_LowDensity_1.txt	BFS	129	32	12,9	3,2	0,022936583	0,003001928
MatrizRandom_75x75_LowDensity_2.txt	BFS	56	110	5,6	11	0,004756927	0,010510921
MatrizRandom_75x75_LowDensity_3.txt	BFS	33	131	3,3	13,1	0	0,013416052
MatrizRandom_75x75_LowDensity_4.txt	BFS	92	76	9,2	7,6	0,010006666	0,009960651
MatrizRandom_75x75_LowDensity_5.txt	BFS	49	121	4,9	12,1	0,004995823	0,01191926
MatrizRandom_75x75_MediumDensity_1.txt	BFS	39	145	3,9	14,5	0,001860857	0,011966705
MatrizRandom_75x75_MediumDensity_2.txt	BFS	85	96	8,5	9,6	0,005496264	0,010042191
MatrizRandom_75x75_MediumDensity_3.txt	BFS	86	92	8,6	9,2	0,005232573	0,013319492
MatrizRandom_75x75_MediumDensity_4.txt	BFS	140	54	14	5,4	0,011074781	0,004004478
MatrizRandom_75x75_MediumDensity_5.txt	BFS	124	50	12,4	5	0,01350975	0,006198168
Metricas	Media	96,07692308	90,30769231	9,607692308	9,030769231	0,007607258	0,007945556
	Mediana	86	96	8,6	9,6	0,005496264	0,009960651
	DesvioPadrão	45,91540975	39,46836048	4,591540975	3,946836048	0,005808848	0,004241197
	Maximo	195	145	19,5	14,5	0,022936583	0,013416052
	Minimo	33	20	3,3	2	0	0,000999689

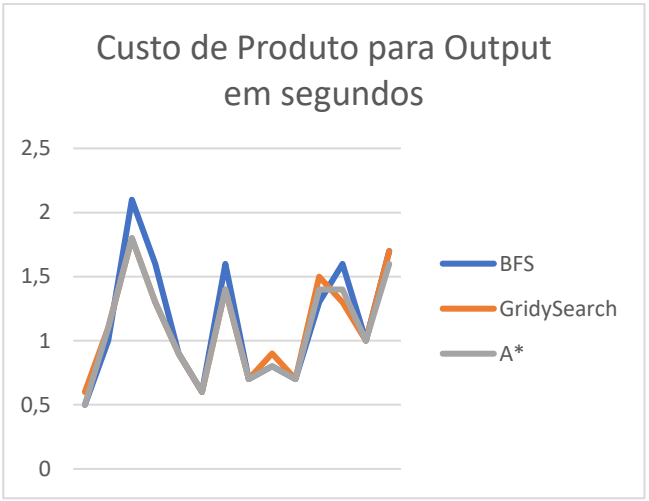
Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_75x75_HighDensity_1.txt	GreedySearch	101	180	10,1	18	0,002024412	0,002145767
MatrizRandom_75x75_HighDensity_2.txt	GreedySearch	215	20	21,5	2	0,004173517	0,001000166
MatrizRandom_75x75_HighDensity_3.txt	GreedySearch	148	127	14,8	12,7	0,007326841	0,00300169
MatrizRandom_75x75_LowDensity_1.txt	GreedySearch	142	32	14,2	3,2	0,005330563	0,000426292
MatrizRandom_75x75_LowDensity_2.txt	GreedySearch	61	110	6,1	11	0,002296448	0,00410223
MatrizRandom_75x75_LowDensity_3.txt	GreedySearch	31	143	3,1	14,3	0,002054453	0,006694078
MatrizRandom_75x75_LowDensity_4.txt	GreedySearch	94	80	9,4	8	0,004138708	0,002001524
MatrizRandom_75x75_LowDensity_5.txt	GreedySearch	47	125	4,7	12,5	0,002556801	0,00420332
MatrizRandom_75x75_MediumDensity_1.txt	GreedySearch	46	163	4,6	16,3	0,002035856	0,00547123
MatrizRandom_75x75_MediumDensity_2.txt	GreedySearch	108	110	10,8	11	0,003434896	0
MatrizRandom_75x75_MediumDensity_3.txt	GreedySearch	111	106	11,1	10,6	0,003149509	0,002776384
MatrizRandom_75x75_MediumDensity_4.txt	GreedySearch	154	45	15,4	4,5	0,001422882	0,00104022
MatrizRandom_75x75_MediumDensity_5.txt	GreedySearch	162	60	16,2	6	0,002448797	0,005186081
Metricas	Media	109,2307692	100,0769231	10,92307692	10,00769231	0,003261053	0,002926845
	Mediana	108	110	10,8	11	0,002556801	0,002776384
	DesvioPadrão	52,01649716	47,92375601	5,201649716	4,792375601	0,001578957	0,00200703
	Maximo	215	180	21,5	18	0,007326841	0,006694078
	Minimo	31	20	3,1	2	0,001422882	0

Matrix Type	Algoritm	Custo de Robo para Produto	Custo de Produto para Output	Custo de Robô para Produto em segundos	Custo de Produto para Output em segundos	Tempo de procura do caminho do Robô para Produto	Tempo de procura do caminho do Produto para Output
MatrizRandom_75x75_HighDensity_1.txt	A*	69	126	6,9	12,6	0,003998041	0,039000273
MatrizRandom_75x75_HighDensity_2.txt	A*	170	19	17	1,9	0,062531471	0
MatrizRandom_75x75_HighDensity_3.txt	A*	133	97	13,3	9,7	0,027179718	0,008229017
MatrizRandom_75x75_LowDensity_1.txt	A*	124	33	12,4	3,3	0,31274724	0,008459806
MatrizRandom_75x75_LowDensity_2.txt	A*	55	104	5,5	10,4	0,007448196	0,070704222
MatrizRandom_75x75_LowDensity_3.txt	A*	31	125	3,1	12,5	0,002977848	0,345920086
MatrizRandom_75x75_LowDensity_4.txt	A*	83	75	8,3	7,5	0,078719854	0,023503304
MatrizRandom_75x75_LowDensity_5.txt	A*	44	115	4,4	11,5	0,006238461	0,216316462
MatrizRandom_75x75_MediumDensity_1.txt	A*	39	127	3,9	12,7	0,003999949	0,143074751
MatrizRandom_75x75_MediumDensity_2.txt	A*	79	86	7,9	8,6	0,015814543	0,033282042
MatrizRandom_75x75_MediumDensity_3.txt	A*	82	83	8,2	8,3	0,030931473	0,054956675
MatrizRandom_75x75_MediumDensity_4.txt	A*	120	43	12	4,3	0,074284554	0,005272627
MatrizRandom_75x75_MediumDensity_5.txt	A*	119	47	11,9	4,7	0,1144104	0,008999348
Metricas	Media	88,30769231	83,07692308	8,830769231	8,307692308	0,057021673	0,073670662
	Mediana	82	86	8,2	8,6	0,027179718	0,033282042
	DesvioPadrão	40,38923051	35,92184152	4,038923051	3,592184152	0,081494447	0,099156696
	Maximo	170	127	17	12,7	0,31274724	0,345920086
	Minimo	31	19	3,1	1,9	0,002977848	0

Figura 11 - Resultados Matrizes 75x75


Custo de Robo para Produto
250
200
150
100
50
0
BFS
GreedySearch
A*
Custo de Produto para Output
250
200
150
100
50
0
BFS
GreedySearch
A*
Tempo de procura do caminho do Robô para Produto
0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
BFS
GreedySearch
A*
Tempo de procura do caminho do Produto para Output
0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0
BFS
GreedySearch
A*





	BFS	*Greedy Search*	A*
Tempo Total Médio (Tempo Movimento do Robô + Tempo de Processamento)	18,65401435	20,93695713	17,26915387

Tabela 6 - Tempo Total Médio de cada Algoritmo em Matrizes 75x75

6. Analise dos Resultados

A análise comparativa permitirá entender a eficácia e a eficiência de cada algoritmo em termos de custo do caminho, tempo de execução e adequação a diferentes densidades de obstáculos.

6.1. Avaliação Geral

- **BFS:**
    - Mostrou-se bastante consistente em todas as dimensões, com tempos de procura geralmente baixos.
    - No entanto, os custos dos caminhos tendem a ser mais altos, indicando que o BFS pode não ser o melhor em termos de otimização de caminho, mas sim em termos de confiabilidade e velocidade.
- ***Greedy Search:***
    - Demonstrou maior variabilidade em custos de caminho e tempo de execução, o que sugere que pode ser sensível à configuração específica da matriz.
    - Em matrizes maiores, o *Greedy Search* apresentou um aumento significativo tanto no custo do caminho quanto no tempo total de execução, possivelmente devido à sua natureza de optar pela escolha localmente ótima sem considerar o custo total do caminho.
- **A*:**
    - Apresentou um equilíbrio notável entre o custo do caminho e o tempo de execução.
    - Foi particularmente eficiente em matrizes de maior densidade, sugerindo que a sua abordagem heurística está bem ajustada para lidar com complexidades adicionais no ambiente.

6.2. Análise Específica por Dimensões

- **Matrizes 10x10:**
    - O A* teve o menor tempo total de execução e apresentou consistentemente bons resultados tanto no custo quanto no tempo, o que sugere uma alta eficiência em espaços pequenos.
    - O BFS mostrou tempos de execução rápidos, mas com custos de caminho ligeiramente maiores.
- **Matrizes 25x25 e 48x52:**
    - A diferença de desempenho entre o BFS e o A* começa a ser mais evidente, com o A* mantendo custos de caminho mais baixos e tempos de execução competitivos mesmo à medida que o tamanho da matriz aumenta.
    - O *Greedy Search* começa a mostrar uma degradação no desempenho, com tempos de execução e custos de caminho cada vez mais altos.
- **Matrizes 50x50 e 75x75:**
    - O A* demonstra uma capacidade impressionante de escalar com o tamanho da matriz, mantendo a eficiência no custo do caminho e no tempo de execução que teve pouca variação.
    - O *BFS*, apesar de manter a confiabilidade, começa a sofrer com o aumento do custo do caminho.
    - O *Greedy Search* revela sua maior desvantagem no aumento do tamanho das matrizes, com tempos de execução baixos, mas custos de caminho mais altos.

7. Pros e Contras

O algoritmo A* mostrou-se o mais versátil e eficiente em todas as dimensões, equilibrando com sucesso o custo do caminho e o tempo de execução. A consistência nos custos de caminho e nos tempos de procura pode indicar robustez em diferentes configurações do armazém.

O algoritmo *BFS* tende a ter um desempenho uniforme e previsível, mas pode não ser o mais eficiente em termos de custo do caminho. Este algoritmo é recomendável para ambientes onde a velocidade é mais crítica do que a otimização do caminho e onde os recursos computacionais são limitados.

O algoritmo *Greedy Search* pode ser útil em cenários onde as soluções aproximadas são aceitáveis. Apresenta uma maior variabilidade nos resultados, indicando uma sensibilidade a configurações específicas.

8. Conclusão

A realização do trabalho mostrou-se, especialmente importante, na medida em que, nos foi possível consolidar os conhecimentos adquiridos na unidade curricular lecionada ao longo do 1º semestre, e adquirir outros, durante o desenvolvimento do trabalho prático acima detalhado.

Relativamente aos pontos propostos para o trabalho prático, consideramos que todos eles foram concretizados com sucesso, com isto referimo-nos ao cumprimento e resolução de todos os pressupostos para a realização do mesmo, incluindo os respetivos testes pedidos assim como o uso das ferramentas necessárias para comprovar os mesmos.

9. Bibliografia

[1] https://github.com/Zav04/AI_PROJECT.git
<consultado a 2-12-2023>

[2] https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/
<consultado a 12-10-2023>

[3] https://www.geeksforgeeks.org/a-search-algorithm/
<consultado a 20-10-2023>

[4] https://www.geeksforgeeks.org/greedy-algorithms/
<consultado a 15-01-2023>