

Estudo do funcionamento e da complexidade computacional do Algoritmo A*



Estrutura de Dados e Algoritmos

Pedro Santos 12551
Emanuel Teixeira 12552

Sumário

- ❖ Introdução
 - ❖ Objetivos e motivação
- ❖ Parte Teórica
 - ❖ Pseudo-código
- ❖ Parte Experimental
 - ❖ Realização Experimental
 - ❖ Sistema Experimental
 - ❖ Análise dos resultados experimentais obtidos
- ❖ Conclusão

Introdução

- ❖ Nesta unidade curricular foi-nos proposto o estudo de um algoritmo de pesquisa heurística em grafos, mais conhecido por A^* .
- ❖ O A^* , ou *pathfinding*, tem como objetivo procurar o caminho com menor custo, de um ponto inicial até um ponto final, utilizando a melhor heurística para o efeito pretendido.

Objetivos e motivação

- ❖ Programação do algoritmo A^* com base em pseudocódigo resultante de pesquisa;
- ❖ Decidimos desenvolver o trabalho em *Python* por ser uma linguagem de alto nível;
 - ❖ Resultado final sucinto e elegante;
- ❖ Utilizámos a biblioteca *OpenCV*, como proposto pelo docente.

Parte Teórica

- ❖ Um algoritmo é uma sequência finita de instruções bem definidas e não ambíguas;
- ❖ Tem o objetivo de resolver problemas, desde que esteja corretamente implementado e adequado ao problema;
- ❖ *Pathfinding* - método de encontrar o caminho com menos custo de um ponto inicial a um ponto final, evitando pontos bloqueados.

Realização Experimental

- ❖ Hardware:

- ❖ MacBook Pro 2.7GHz;

- ❖ MacBook Pro 2.5GHz;

- ❖ Software:

- ❖ OSX 10.9 Mavericks;

- ❖ Debian 7.1;

- ❖ *Sublime Text*;

- ❖ *Python*;

- ❖ *VirtualBox*;

Sistema Experimental

❖ Este Projeto contém dois módulos:

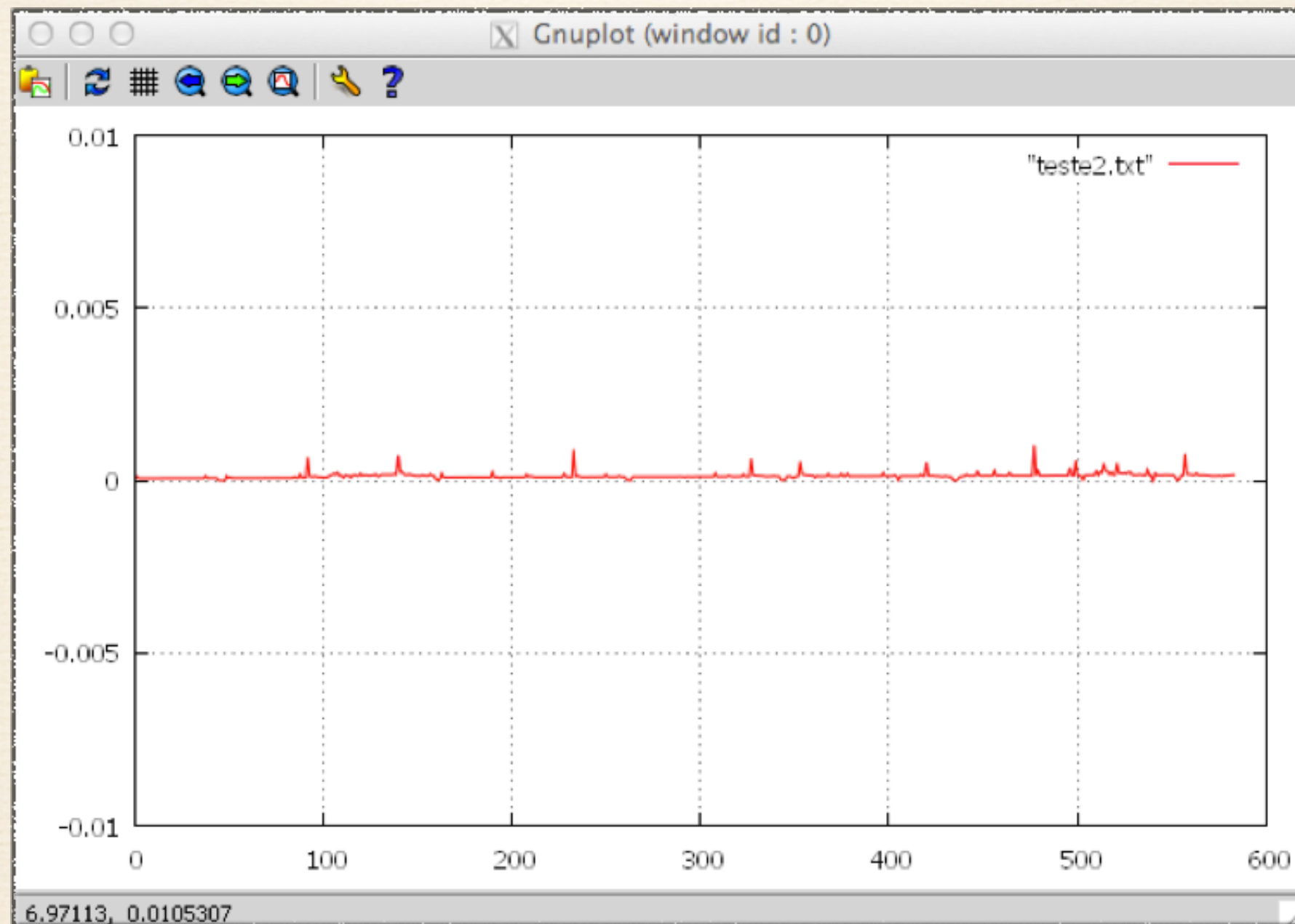
❖ main.py

❖ astar.py

```
astar.py
1  import math
2
3  '''
4  @authors 12551 Pedro Santos & 12552 Emanuel Teixeira
5  @date 11 de Maio de 2014
6  @obs: Algoritmo A* Path Finding para a
7  procura de caminhos com menos custo
8  em funcao da intensidade.
9  '''
10
11 class AStar():
12     """Classe AStar e responsavel pelo algoritmo da busca
13     do melhor caminho com menos custo"""
14     def __init__(self):
15         '''
16         Inicializacao dos parametros da classe Astar
17         '''
18         self.START_POINT = (191, 48) #x, y
19         self.END_POINT = (260, 508)
20         self.ALPHA = 1
21         self.MAX_COR = 255.0
22         self.closed_set = []
23
24     def g(self, neighbor):
```

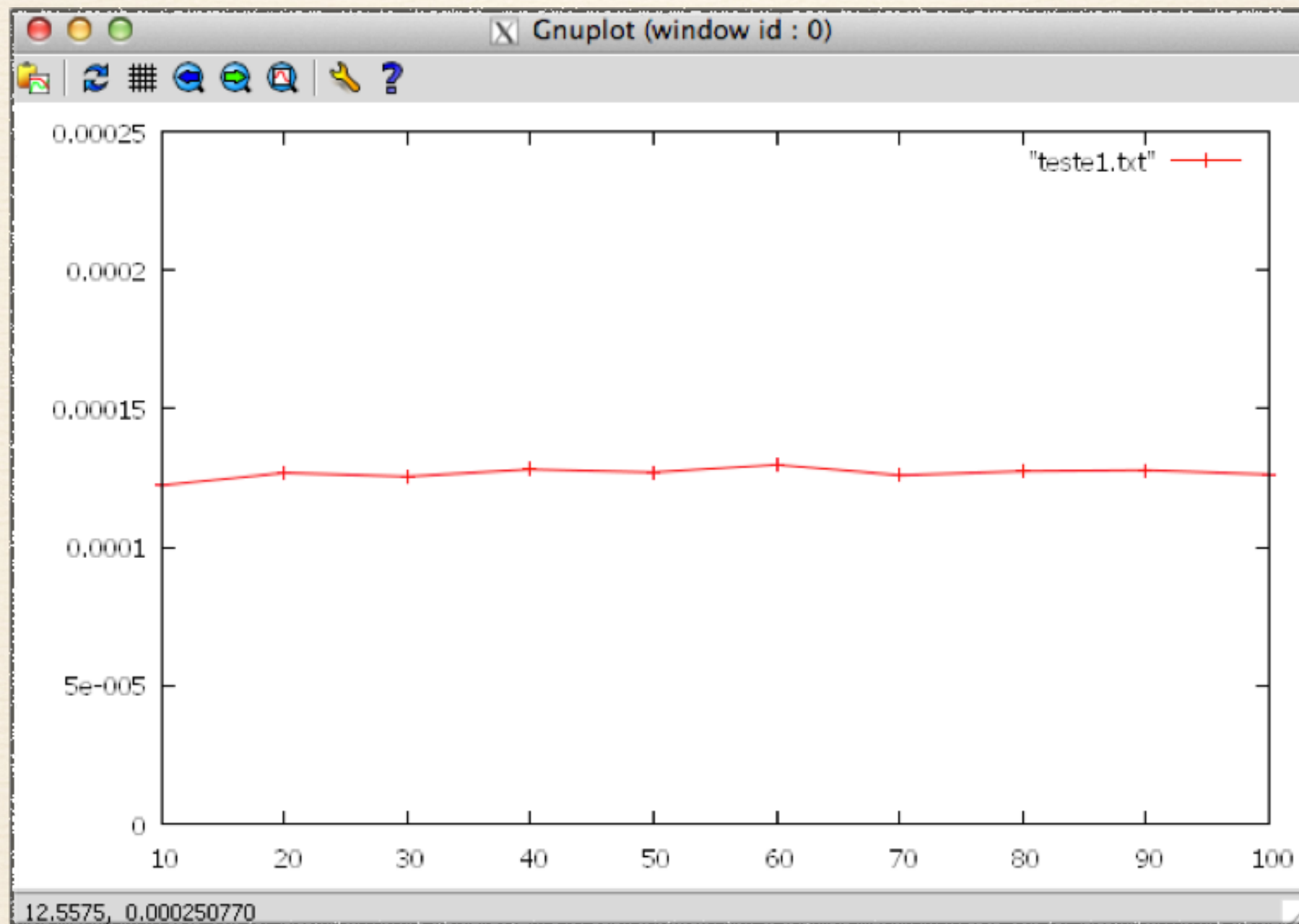

Análise dos resultados experimentais obtidos

- ❖ Tempo necessário para calcular um novo ponto com o custo mais reduzido (cerca de 580 pontos):



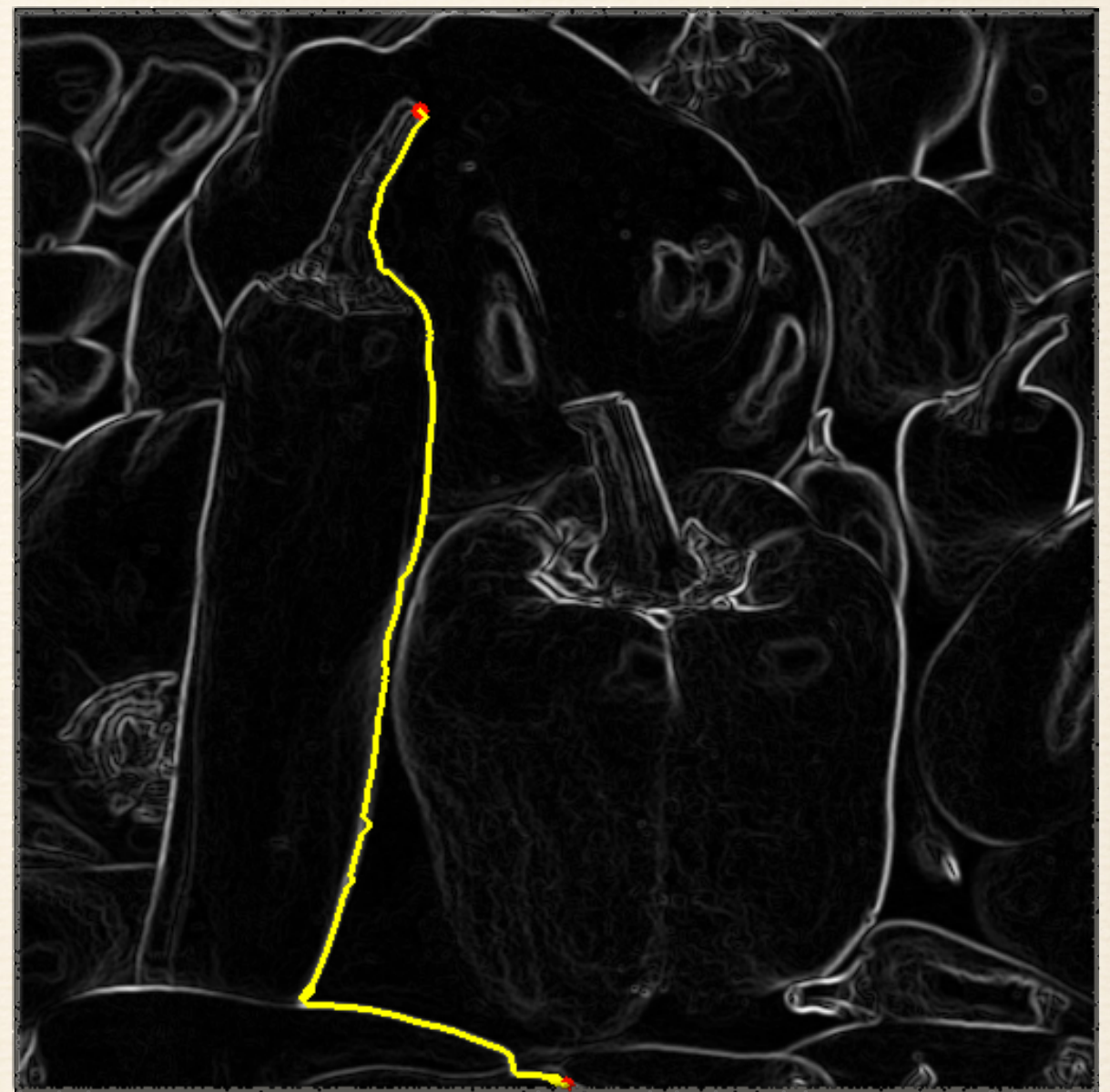
Análise dos resultados experimentais obtidos

- ❖ Média de tempos de procura de pontos por execução e de seguida a média entre todas as execuções. O gráfico mantém-se linear.



Análise dos resultados experimentais obtidos

❖ Imagem original e o resultado final, respectivamente:



Conclusão

- ❖ Objetivos propostos pelo docente concluídos com sucesso;
- ❖ Aperfeiçoamento dos conhecimentos em *Python*;
- ❖ O Algoritmo A* apresenta uma complexidade computacional de $O(n)$ para cada ponto da sua busca;
- ❖ Código desenvolvido utilizando o mínimo de operações possível de maneira a tornar o algoritmo mais eficiente;
- ❖ O resultado foi o esperado, indo pelo caminho com menos custo;
- ❖ Aplicação totalmente funcional.