t

Licenciatura em Engenharia Informática

# Índice

[Índice 3](#_Toc93265469)

[Introdução 4](#_Toc93265470)

[Implementação 5](#_Toc93265471)

[Trepa-Colinas (com vizinhança 1) 5](#_Toc93265472)

[Trepa-Colinas (probabilístico) 7](#_Toc93265473)

[Torneio Binário 8](#_Toc93265474)

[Análise de Resultados 11](#_Toc93265475)

[Trepa-Colinas (com vizinhança 1) 11](#_Toc93265476)

[Não aceitando soluções iguais 11](#_Toc93265477)

[Aceitando soluções iguais 11](#_Toc93265478)

[Trepa-Colinas (probabilístico) 12](#_Toc93265479)

[Não aceitando soluções iguais 12](#_Toc93265480)

[Aceitando soluções iguais 12](#_Toc93265481)

[Conclusão 13](#_Toc93265482)

# Introdução

O segundo trabalho prático de Introdução à Inteligência Artificial consiste no desenvolvimento de um programa na linguagem de programação C capaz de resolver o Problema do Conjunto Estável Máximo.

Foram-nos fornecidas várias instâncias de variados tamanhos para corrermos no programa e efetuarmos um estudo experimental.

Uma imagem com texto

Descrição gerada automaticamente

# Implementação

## Trepa-Colinas (com vizinhança 1)

Começamos por ler o ficheiro contendo a informação das ligações entre os vértices com a função preenche\_matriz e seguidamente alocamos memória para os vetores que vão guardar a solução atual e a melhor solução.

Uma imagem com texto

Descrição gerada automaticamente

Para efetuarmos várias iterações do Trepa-Colinas corremos um ciclo o número de vezes definida anteriormente que consiste em gerar uma solução com o trepa-colinas utilizando a solução anterior e verificando se a solução devolvida pelo algoritmo é superior à anterior (problema de maximização). Caso isso se verifique a melhor solução é substituída com a solução calculada.

Uma imagem com texto

Descrição gerada automaticamente

No final do ciclo, é mostrada a melhor solução e a média das soluções encontradas para o ficheiro que foi dado ao programa no início.

Uma imagem com texto

Descrição gerada automaticamente

O trepa-colinas com vizinhança 1, avalia a solução, e aceita o vizinho apenas se este for maior do que o atual.

Uma imagem com texto

Descrição gerada automaticamente

## Trepa-Colinas (probabilístico)

O trepa-colinas probabilístico é bastante semelhante ao trepa-colinas só que para além de depender de um vizinho depende de uma probabilidade pré-determinada.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente Uma imagem com texto, ecrã, captura de ecrã

Descrição gerada automaticamente

## Torneio Binário

O algoritmo Torneio Binário, ou em inglês, Binary Tournament foi implementado da seguinte forma:

Uma imagem com texto

Descrição gerada automaticamente Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Na função main, temos um ciclo para correr o algoritmo n vezes de modo a aprimorar a solução final.

Uma imagem com texto

Descrição gerada automaticamente

No final do ciclo é mostrada a solução final com maior cardinalidade (problema de maximização).

# Análise de Resultados

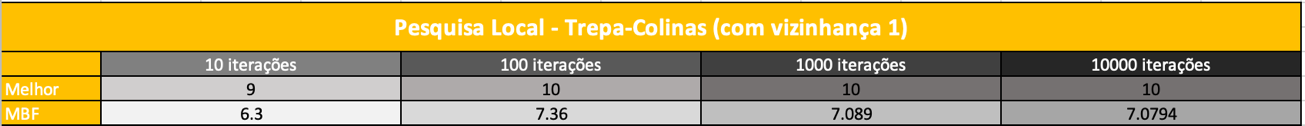
O estudo experimental consistiu em correr o programa com vários ficheiros e com um número de iterações variável para observar a qualidade das soluções calculadas.

Como é de esperar, correr 10000 vezes o algoritmo leva-nos mais perto da solução.

## Trepa-Colinas (com vizinhança 1)

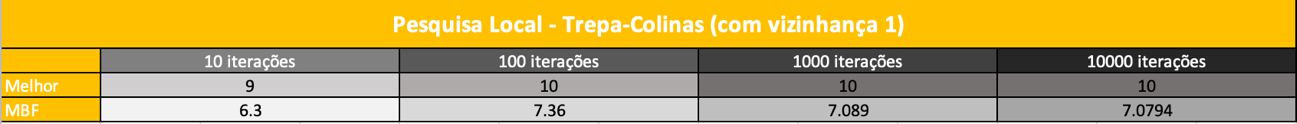
### Uma imagem com texto Descrição gerada automaticamenteNão aceitando soluções iguais

Para não aceitar soluções iguais basta verificar se o custa da solução calculada é maior do que o custo da melhor solução. Se isso se verificar basta substituir a melhor solução com a solução calculada.



### Uma imagem com texto Descrição gerada automaticamenteAceitando soluções iguais

De forma a aceitar soluções iguais, quando verificamos se uma solução calculada é melhor do que a melhor atual, basta verificar se ela é maior ou **igual**. Desta forma, aceitamos soluções que são tão boas ou melhores do que a melhor global.



## Trepa-Colinas (probabilístico)

### Uma imagem com texto Descrição gerada automaticamenteNão aceitando soluções iguais

Para não aceitar soluções iguais basta verificar se o custa da solução calculada é maior do que o custo da melhor solução. Se isso se verificar basta substituir a melhor solução com a solução calculada.

Uma imagem com texto, captura de ecrã, monitor, ecrã

Descrição gerada automaticamente

Foram testadas diferentes probabilidades.

### Aceitando soluções iguais

Uma imagem com texto

Descrição gerada automaticamenteDe forma a aceitar soluções iguais, quando verificamos se uma solução calculada é melhor do que a melhor atual, basta verificar se ela é maior ou **igual**. Desta forma, aceitamos soluções que são tão boas ou melhores do que a melhor global.

Uma imagem com texto, captura de ecrã, ecrã, prateado

Descrição gerada automaticamente

Foram testadas diferentes probabilidades.

# Conclusão

Este trabalho permitiu-nos observar vários algoritmos em funcionamento capazes de resolver um problema utilizando aprendizagem contínua. Para além disso permitiu-nos aprimorar o nosso nível de programação em C e tratamento de dados.

O estudo experimental permitiu-nos verificar os resultados obtidos e o quanto eles variavam com a mudança de certos parâmetros.

