Trabalho Prático nº 2

Problema de Otimização Max Subgrafo

Francisco José Valente Ferreira

2019113494

Licenciatura em Engenharia Informática

Ramo de Desenvolvimento de Aplicações

Instituto Politécnico de Coimbra

Instituto Superior de Engenharia de Coimbra

09, janeiro de 2024

**Índice**

[Índice de figuras v](#_Toc123572065)

[Índice de tabelas vii](#_Toc123572066)

[1 Introdução 1](#_Toc123572067)

[1.1 Estrutura do relatório 1](#_Toc123572068)

[2 representação do problema 3](#_Toc123572069)

[2.1 Modelo de dados 3](#_Toc123572070)

[3 Métodos de otimização 5](#_Toc123572071)

[3.1 Trepa colinas first choice 5](#_Toc123572072)

[3.1.1 Vizinhança 1 5](#_Toc123572073)

[3.1.2 Vizinhança 2 5](#_Toc123572074)

[3.1.3 Vizinhança 3 5](#_Toc123572075)

[3.1.4 Vizinhança 4 6](#_Toc123572076)

[3.2 Algoritmo genético 6](#_Toc123572077)

[3.2.1 Recombinação de ponto único 6](#_Toc123572078)

[3.2.2 Recombinação de dois pontos 6](#_Toc123572079)

[3.2.3 Recombinação uniforme 6](#_Toc123572080)

[4 experimentação e análise 7](#_Toc123572081)

[4.1 Trepa colinas 7](#_Toc123572082)

[4.1.1 Análise de aceitação de maiores custos ou custos iguais. 8](#_Toc123572083)

[4.1.2 Análise de vizinhos 9](#_Toc123572084)

[4.2 Algoritmo genético 10](#_Toc123572085)

[4.2.1 Análise de probabilidade de recombinação 11](#_Toc123572086)

[4.2.2 Análise de probabilidade de mutação 11](#_Toc123572087)

[4.2.3 Análise do tamanho da população 11](#_Toc123572088)

[4.2.4 Análise das 3 recombinações 12](#_Toc123572089)

[4.3 Algoritmo Híbrido 13](#_Toc123572090)

[4.3.1 Análise de genético com trepa colina na população inicial (Híbrido 1) 13](#_Toc123572091)

[4.3.2 Análise de genético com trepa colina no último elemento melhor 13](#_Toc123572092)

[4.3.3 Otimização final com trepa colinas no início e fim 14](#_Toc123572093)

[5 Melhor 15](#_Toc123572094)

[6 Conclusões 17](#_Toc123572095)

# Índice de figuras

[Figura 1 Estrutura dos ficheiros, Matriz de adjacências, Desenho do grafo. 3](#_Toc123572059)

[Figura 2 Exemplos de soluções válidas para k = 4 3](#_Toc123572060)

# Índice de tabelas

[Tabela 1 trepa colinas 100,200,500,1000 7](#_Toc123572047)

[Tabela 2 Aceitar maior, aceitar maior ou igual 8](#_Toc123572048)

[Tabela 3 Vizinho 1 e Vizinho 2 9](#_Toc123572049)

[Tabela 4 Vizinho 3, Vizinho4 9](#_Toc123572050)

[Tabela 5 Variação Genético 100, 200 10](#_Toc123572051)

[Tabela 6 Variação Genético 500, 1000 10](#_Toc123572052)

[Tabela 7 Genético, aumento recombinação 11](#_Toc123572053)

[Tabela 8 Genético, aumento mutação 11](#_Toc123572054)

[Tabela 9 Genético, aumento população 11](#_Toc123572055)

[Tabela 10 Genético, 1 ponto, 2 pontos, uniforme, cada 500 iterações 12](#_Toc123572056)

[Tabela 11 Híbrido, Genético VS Híbrido 1 13](#_Toc123572057)

[Tabela 12 Híbrido 2 13](#_Toc123572058)

# Introdução

No contexto da disciplina de Inteligência Artificial o projeto consistiu em Implementar em código C algoritmos de otimização, usando a trepa colinas, e variações de algoritmos genéticos em específico o problema de maximização de arestas de um subgrafo.

Neste trabalho foram implementados 3 algoritmos, o trepa colina, um algoritmo genético, e um algoritmo de hibrido, sendo alteradas configurações para obter os melhores valores de otimização.

O objetivo foi efetuar um estudo comparativo aprofundado sobre o desempenho da otimização, usando exemplos disponibilizados de um pequeno conjunto de vértices até elevados valores e testar a otimização entre os diferentes ficheiros.

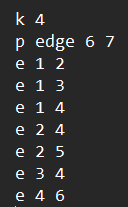
## Estrutura do relatório

O relatório vai falar sobre a representação do problema e abordagens para gerar soluções válidas, seguindo das configurações e resultados para o algoritmo de pesquisa local, configurações e resultados para o algoritmo evolutivo, configurações e resultados para os algoritmos híbridos.

# representação do problema

## Modelo de dados

Os valores dos dados são obtidos através ficheiros com estrutura pré-definida. São obtidos os valores do número de itens do subconjunto de vértices (k), o número vértices p = 6 e o número de arestas. A matriz de adjacências p\*p (Figura 1)é criada, onde são colocados 1s onde há ligações, só é preenchida a parte superior ou inferior, e a diagonal está preenchida a 0 senão significava que um ponto está ligado com ele mesmo e contar como aumento do custo.

 Uma imagem com texto, eletrónica, calculadora, teclado

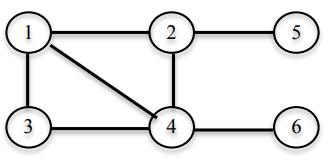
Descrição gerada automaticamente 

Figura Estrutura dos ficheiros, Matriz de adjacências, Desenho do grafo.

A representação de dados consiste num vetor de p bits em que 1 representa que faz parte do subconjunto k, 0 representa que não está. A fitness é a soma das arestas entre todos os elementos 1 desse vetor, de forma a obter mais arestas possíveis, sendo este um problema de maximização do fitness. Uma solução inválida existe quando não é respeitado o número k de elementos num grupo, ao existir mais ou menos que k.

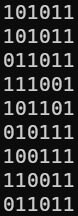


Figura Exemplos de soluções válidas para k = 4

Para obter a fitness da solução vai se a cada elemento do subconjunto k, ou seja, todos os elementos marcados com 1, de seguida procurasse outro elemento marcado com 1 e para a posição i, j dos referidos procurasse na matriz de adjacências que neste caso é um vetor 1d, *row major*.

# Métodos de otimização

## Trepa colinas first choice

O trepa colinas é um algoritmo de otimização heurístico que é utilizado para encontrar o máximo local de uma função. Que vai criando alterações iterativas na solução atual e escolhendo a alteração que resulta em maior fitness na solução.

A variante "first choice" de trepa colinas é um tipo específico de algoritmo de trepa colinas que considera apenas a primeira possível mudança que melhora a solução, em vez de considerar todas as mudanças possíveis e escolher a melhor. Isso pode ser uma estratégia útil quando o espaço de busca é grande e o custo de avaliar as mudanças potenciais é alto, pois permite que o algoritmo progrida mais rapidamente. No entanto, também é mais propenso a ficar preso em máximos locais, pois pode não considerar mudanças alternativas que poderiam potencialmente levar a uma solução melhor.

Para ilustrar como o trepa colinas funciona, exemplo: suponha que desejamos encontrar o valor máximo da função f(x) = x^2 no intervalo [0, 10]. Podemos começar escolhendo um ponto inicial, digamos x = 5, e então fazer alterações iterativas em x para ver se podemos melhorar o valor da função. Se aumentarmos x em 1, encontramos que f(6) = 36, o que é maior que f(5) = 25. Portanto, atualizamos nossa solução atual para x = 6. Em seguida, repetimos o processo, aumentando x em 1 novamente para encontrar que f(7) = 49, o que é maior que f(6). Continuamos esse processo até atingirmos o valor máximo da função em x = 10.

O algoritmo só termina quando o número de iterações especificado termina. Guardando o último melhor fitness que será a solução ótima.

Neste trabalho o algoritmo é lançado várias vezes em diversos pontos aleatórios, tentando assim encontrar ótimos globais.

### Vizinhança 1

A vizinhança um escolhe um valor aleatório numa posição 0 e procura outro valor aleatório numa posição 1 e troca os valores.

### Vizinhança 2

A vizinhança um escolhe um valor aleatório e se o elemento logo a sua direita for diferente troca

### Vizinhança 3

A vizinhança 3 faz uma troca como a vizinhança dois mais envés de ser logo o elemento logo a sua direita é o elemento logo a seguir, ou seja, i + 2.

### Vizinhança 4

A vizinhança 4 escolhe duas posições, uma de 0 até ao meio do vetor e outra acima dessa metade, e inverte esse subconjunto, ou seja, faz um reverse do subconjunto selecionado.

## Algoritmo genético

O algoritmo genético é composto por 3 fases. A Recombinação genética é o processo de misturar os genes de dois indivíduos diferentes para criar um novo indivíduo com características combinadas de ambos os pais. Isso ocorre naturalmente em seres vivos.

Mutação é o processo de alteração aleatória de um gene em um indivíduo. As mutações podem resultar em características novas ou inesperadas no indivíduo.

Seleção é o processo de escolha de indivíduos com base em suas características para reprodução. Em algoritmos genéticos, isso geralmente é feito de acordo com a aptidão do indivíduo, ou seja, sua capacidade de se adaptar e sobreviver em um determinado ambiente. Indivíduos com características mais desejáveis são mais propensos a serem selecionados.

### Recombinação de ponto único

Envolve a seleção de um único ponto no material genético das duas soluções parentais e a troca do material genético em ambos os lados desse ponto.

### Recombinação de dois pontos

Cruzamento de dois pontos: semelhante ao cruzamento de ponto único, mas envolve a seleção de dois pontos no material genético das duas soluções parentais e a troca do material entre os dois pontos para criar a descendência.

### Recombinação uniforme

Cruzamento uniforme: envolve a seleção aleatória de genes individuais de qualquer um dos pais para criar a prole. Isso pode resultar em descendentes mais semelhantes a um dos pais ou a outro, ou que são uma mistura dos dois.

# experimentação e análise

## Trepa colinas

Na experimentação e análise do trepa colinas inicial, foram feitas para 100, 200, 500, 1000 iterações, com configuração em que aceita apenas maiores, com o vizinho 1, ou seja aleatório trocas entre 1 e 0.

É possível ver o aumento do custo com o número de iterações, mas a estabilização de da média e até inicio de declínio em maiores iterações em ficheiros mais pequenos (file2, file3).

Tabela trepa colinas 100,200,500,1000

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 100 | Média | Custo | 200 | Média | Custo |
| teste.txt | 5.000000 | 5 | teste.txt | 5.000000 | 5 |
| file1.txt | 19.660000 | 20 | file1.txt | 19.684999 | 20 |
| file2.txt | 13.910000 | 15 | file2.txt | 13.980000 | 15 |
| file3.txt | 105.040001 | 108 | file3.txt | 104.970001 | 110 |
| file4.txt | 31.709999 | 46 | file4.txt | 30.795000 | 46 |
| file5.txt | 20.629999 | 32 | file5.txt | 20.379999 | 32 |
|  | | | | | |
| 500 | Média | Custo | 1000 | Média | Custo |
| teste.txt | 5.000000 | 5 | teste.txt | 5.000000 | 5 |
| file1.txt | 19.730000 | 20 | file1.txt | 19.731001 | 20 |
| file2.txt | 14.116000 | 15 | file2.txt | 14.067000 | 15 |
| file3.txt | 104.942001 | 111 | file3.txt | 104.893997 | 111 |
| file4.txt | 30.782000 | 50 | file4.txt | 30.348000 | 54 |
| file5.txt | 20.686001 | 35 | file5.txt | 20.760000 | 38 |

### Análise de aceitação de maiores custos ou custos iguais.

Análise de diferença entre aceitar maior ou aceitar maior ou igual para 500 iterações. Permitiu verificar o impacto de aceitar soluções iguais o que aumenta as médias. Porém para um determinado número de iterações nem sempre dá o melhor resultado geral pois pode seguir outro caminho que não faria se aceitasse soluções iguais.

Tabela Aceitar maior, aceitar maior ou igual

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 500 | Média | Custo | 500 | Média | Custo |
| teste.txt | 5.000000 | 5 | teste.txt | 5.000000 | 5 |
| file1.txt | 19.730000 | 20 | file1.txt | 19.784000 | 20 |
| file2.txt | 14.116000 | 15 | file2.txt | 14.520000 | 15 |
| file3.txt | 104.942001 | 111 | file3.txt | 105.269997 | 110 |
| file4.txt | 30.782000 | 50 | file4.txt | 30.521999 | 56 |
| file5.txt | 20.686001 | 35 | file5.txt | 20.856001 | 38 |

### Análise de vizinhos

Na análise de vizinhos foi possível análise 4 algoritmos para 500 iterações cada ficheiro, aceitando soluções maiores ou igual, ver que tipo de alterações provoca mais qualidade nas soluções geradas. O vizinho 1 foi obtido nas aulas, que apenas encontra 2 números diferentes e troca o que permite diferenças possivelmente mais relevantes. O vizinho 2 foi o que obteve piores resultados, a diferença colocada no algoritmo é que apenas troca 2 valores diferentes adjacentes o que pode levar a trocas pouco significativas para o mesmo número de iterações.

Tabela Vizinho 1 e Vizinho 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Média | Custo | 2 | Média | Custo |
| teste.txt | 5.000000 | 5 | teste.txt | 3.890000 | 5 |
| file1.txt | 19.784000 | 20 | file1.txt | 18.840000 | 20 |
| file2.txt | 14.520000 | 15 | file2.txt | 12.536000 | 15 |
| file3.txt | 105.269997 | 110 | file3.txt | 102.209999 | 108 |
| file4.txt | 30.521999 | 56 | file4.txt | 16.332001 | 40 |
| file5.txt | 20.856001 | 38 | file5.txt | 9.160000 | 23 |

O vizinho 3 aumentou a distância do vizinho 2, ou seja a troca de dois elementos, mas envés de diretamente adjacentes foi incrementado um bit para a direita, o que melhorou notoriamente a qualidade de resultados da média. O vizinho 4 faz um reverse de um ponto da metade inferior até um ponto da metade superior, o pode aumentar a diferença das soluções e por vezes escapar soluções pois faz alterações muito significativas, para ficheiros maiores teve mais dificuldade.

Tabela Vizinho 3, Vizinho4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3 | Média | Custo | 4 | Média | Custo |
| teste.txt | 4.164000 | 5 | teste.txt | 4.918000 | 5 |
| file1.txt | 19.058001 | 20 | file1.txt | 19.242001 | 20 |
| file2.txt | 12.116000 | 15 | file2.txt | 12.640000 | 15 |
| file3.txt | 102.671997 | 108 | file3.txt | 101.585999 | 106 |
| file4.txt | 18.548000 | 36 | file4.txt | 19.488001 | 34 |
| file5.txt | 11.490000 | 32 | file5.txt | 14.128000 | 30 |

## Algoritmo genético

O algoritmo genético foi configurado com uma população inicial de 10 elementos escolhidos todos inicialmente a partir de soluções validas, com um máximo de 100 gerações com probabilidade de mutação de 1%, probabilidade de recombinação de 10%, e inicialmente são selecionados os parentes por torneio de 2. Foram feitas alterações singulares para cada quadro de dados começando das configurações *default* no projeto.

Foi analisada o aumento das iterações no algoritmo genético. Em geral as médias aumentam e o melhores fitness são encontrados para os ficheiros mais pequenos. Para obter resultados muito melhores, só com um grande número de iterações. Sem dúvida que o trepa colinas obtém muito melhores resultados com uma configuração inicial simples.

Tabela Variação Genético 100, 200

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 100 | Média | Custo | 200 | Média | Custo |
| file1.txt | 19.110001 | 20.000000 | file1.txt | 19.080000 | 20.000000 |
| file2.txt | 10.660000 | 14.000000 | file2.txt | 10.685000 | 14.000000 |
| file3.txt | 99.139999 | 103.000000 | file3.txt | 99.195000 | 105.000000 |
| file4.txt | 10.470000 | 15.000000 | file4.txt | 10.760000 | 19.000000 |
| file5.txt | 7.040000 | 11.000000 | file5.txt | 7.045000 | 11.000000 |
| teste.txt | 4.410000 | 5.000000 | teste.txt | 4.485000 | 5.000000 |

Tabela Variação Genético 500, 1000

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 500 | Média | Custo | 1000 | Média | Custo |
| file1.txt | 19.076000 | 20.000000 | file1.txt | 19.063999 | 20.000000 |
| file2.txt | 10.682000 | 14.000000 | file2.txt | 10.748000 | 15.000000 |
| file3.txt | 99.129997 | 105.000000 | file3.txt | 99.223000 | 105.000000 |
| file4.txt | 11.076000 | 19.000000 | file4.txt | 11.182000 | 19.000000 |
| file5.txt | 6.952000 | 14.000000 | file5.txt | 6.902000 | 14.000000 |
| teste.txt | 4.520000 | 5.000000 | teste.txt | 4.539000 | 5.000000 |

### Análise de probabilidade de recombinação

O aumento da probabilidade de recombinação aumentou a qualidade de resultados, a probabilidade de 50% mostrou se relevante na obtenção de melhores médias para todos os ficheiros.

Tabela Genético, aumento recombinação

|  |  |  |
| --- | --- | --- |
| 500 | Média | Custo |
| file1.txt | 19.106001 | 20.000000 |
| file2.txt | 10.866000 | 15.000000 |
| file3.txt | 99.136002 | 105.000000 |
| file4.txt | 11.334000 | 20.000000 |
| file5.txt | 7.014000 | 16.000000 |
| teste.txt | 4.532000 | 5.000000 |

### Análise de probabilidade de mutação

O aumento da probabilidade de mutação de 0.01 para 0.2, provou ser ineficaz fazendo descer as médias significativamente.

Tabela Genético, aumento mutação

|  |  |  |
| --- | --- | --- |
| 500 | Média | Custo |
| file1.txt | 18.618000 | 20.000000 |
| file2.txt | 10.190000 | 14.000000 |
| file3.txt | 97.300003 | 104.000000 |
| file4.txt | 10.968000 | 19.000000 |
| file5.txt | 6.938000 | 14.000000 |
| teste.txt | 5.000000 | 5.000000 |

### Análise do tamanho da população

O aumento da população de 10 para 30, provou ser fundamental para aumentar médias e aumentar o fitness pretendido.

Tabela Genético, aumento população

|  |  |  |
| --- | --- | --- |
| 500 | Média | Custo |
| file1.txt | 19.548000 | 20.000000 |
| file2.txt | 12.074000 | 15.000000 |
| file3.txt | 101.746002 | 106.000000 |
| file4.txt | 13.192000 | 22.000000 |
| file5.txt | 8.248000 | 15.000000 |
| teste.txt | 4.914000 | 5.000000 |

### Análise das 3 recombinações

A análise das combinações de 1, 2 pontos, e uniforme, mostrou que para alguns ficheiros um algoritmo pode ser melhor e para outros pode ser pior. Conclui-se que para este problema convém testar as 3 para melhor certeza de que estamos a obter os melhores resultados, partindo das configurações iniciais.

Tabela Genético, 1 ponto, 2 pontos, uniforme, cada 500 iterações

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 500 – 1 p | Média | Custo | 500 – 2 p | Média | Custo | 500 - uni | Média | Custo |
| file1.txt | 19.076000 | 20 | file1.txt | 19.032000 | 20 | file1.txt | 19.016001 | 20 |
| file2.txt | 10.682000 | 14 | file2.txt | 10.810000 | 15 | file2.txt | 10.764000 | 15 |
| file3.txt | 99.129997 | 105 | file3.txt | 99.244003 | 105 | file3.txt | 99.113998 | 105 |
| file4.txt | 11.076000 | 19 | file4.txt | 11.144000 | 21 | file4.txt | 11.058000 | 22 |
| file5.txt | 6.952000 | 14 | file5.txt | 6.816000 | 16 | file5.txt | 6.946000 | 14 |
| teste.txt | 4.520000 | 5 | teste.txt | 4.502000 | 5 | teste.txt | 4.528000 | 5 |

## Algoritmo Híbrido

A última configuração testada foi a mistura de trepa colinas no início do algoritmo genético e no final, no início o trepa colinas é usado para gerar uma população inicial, e a partir de ai criar iterações no algoritmo genético para obter os melhores resultados a partir de resultados bons. O uso da trepa colinas no final também consegue ter resultados bons pois também vai otimizar a última melhor solução.

As configurações são iguais as anteriores no algoritmo genético, com 500 iterações do algoritmo genético com 100 gerações, recombinação 1, PR 0.1, PM 0.01, POPSIZE 10, o trepa colinas corre com 500 iterações, vizinho 1.

### Análise de genético com trepa colina na população inicial (Híbrido 1)

Ao analisar os ficheiros mais pequenos conseguimos perceber que as médias estão no ponto máximo, é possível ver o impacto de uma população inicial já bastante forte e com uma configuração inicial simples do algoritmo genético.

Tabela Híbrido, Genético VS Híbrido 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Genético simples | Média | Custo | Trepa colinas no final | Média | Custo |
| file1.txt | 19.076000 | 20 | file1.txt | 20 | 20 |
| file2.txt | 10.682000 | 14 | file2.txt | 15 | 15 |
| file3.txt | 99.129997 | 105 | file3.txt | 111.972000 | 112 |
| file4.txt | 11.076000 | 19 | file4.txt | 72.585999 | 79 |
| file5.txt | 6.952000 | 14 | file5.txt | 64.818001 | 90 |
| teste.txt | 4.520000 | 5 | teste.txt | 5 | 5 |

### Análise de genético com trepa colina no último elemento melhor

Em comparação com o algoritmo genético simples, é possível verificar otimização, mas não tão grande como a variação anterior.

Tabela Híbrido 2

|  |  |  |
| --- | --- | --- |
| Trepa colinas no final | Média | Custo |
| file1.txt | 19.025999 | 20 |
| file2.txt | 10.720000 | 15 |
| file3.txt | 99.248001 | 106 |
| file4.txt | 11.386000 | 22 |
| file5.txt | 6.964000 | 14 |
| teste.txt | 4.526000 | 5 |

### Otimização final com trepa colinas no início e fim

Aqui seria onde os resultados seriam os melhores, pois a combinação dos dois só pode trazer melhores resultados. Pois visto que houve melhores em ambos separados.

Como tal houve aumento nas médias e nos resultados em geral dos maiores ficheiros, visto que os mais pequenos (teste, file1, file2, file3) já tinham atingido o ponto máximo de otimização.

|  |  |  |
| --- | --- | --- |
| Otimização inicial e final | Média | Custo |
| file1.txt | 20 | 20 |
| file2.txt | 15 | 15 |
| file3.txt | 111.958000 | 112 |
| file4.txt | 72.907997 | 79 |
| file5.txt | 65.000000 | 87 |
| teste.txt | 5.000000 | 5 |

# Melhor

Os melhores resultados foram obtidos otimizando os parâmetros que previamente foram considerados bons melhoramentos tanto no genético como no trepa colinas, e partindo do algoritmo hibrido final.

Foram aumentados a probabilidade de recombinação e tamanho da população, para valores altos visto que serão dos mais importantes, foram também aumentadas as iterações de forma geral o que aumentou o tempo de processamento.

[hibrid - config 3 ]

[1] Best

PM:0.06

PR:0.8

POPSIZE: 100

MAX\_GEN: 200,

genetic\_runs:700,

inner\_trepa: 700

Média Custo

file1.txt 20.000000 20

file2.txt 15.000000 15

file3.txt 112.000000 112

file4.txt 78.978569 79

file5.txt 83.481430 98

teste.txt 5.000000 5

# Conclusões

Neste trabalho possível consolidar conceitos de algoritmos de trepa colinas bem como algoritmos genéticos , incluindo manipulação de estruturas de dados tendo em conta complexidades temporais bem como espaciais .Permitiu também aprofundar conhecimentos em C, para otimizar a velocidade de processamento dos vários ficheiros avaliado.