Trabalho Prático 2
IIA
Relatório

Feito por:
- Pedro Amorim 2022157609
- Rodrigo Matos 2022143087

Índice:

Objetivo

O objetivo deste trabalho é encontrar um subconjunto de custo mínimo em um determinado grafo.

Para isso usámos 3 métodos:

1. Algoritmo de pesquisa local (trepa colinas)
2. Algoritmo evolutivo
3. Algoritmo Híbrido (junção dos dois algoritmos)

Problema Apresentado

Dado um grafo e um valor inteiro k, este problema consiste em encontrar um subconjunto de vértices de tamanho k, tal que todos os vértices possuam pelo menos uma ligação e o custo das arestas dentro do subconjunto seja mínimo.

Formalmente o problema é definido como um grafo não direcionado $G = (V, A)$, composto por um conjunto V de vértices ligados entre si por arestas pesadas A e um inteiro k.
Dado este grafo o objetivo é encontrar um subconjunto de vértices S, de tamanho k, tal que $S \subseteq V$, de forma a minimizar o custo total das arestas desse subconjunto.

Interface do Programa

```
Ficheiro a ler:test
 Valor K: 4
Num Vertices: 6
Ligacoes: 7
1 - Trepa Colinas
2 - Algoritmo Evolutivo
3 - Algoritmo Hibrido
Escolha:1
 Quantas runs deseja fazer:10
 Usar vizinhanca 1 ou 2:1
 Quantas iteracoes deseja fazer:1000
```

Figura 1 - Algoritmo de pesquisa local (trepa-colinas)

```
Ficheiro a ler:test
 Valor K: 4
Num Vertices: 6
Ligacoes: 7
1 - Trepa Colinas
2 - Algoritmo Evolutivo
3 - Algoritmo Hibrido
Escolha:2
 Quantas runs deseja fazer:10
 Qual o tamanho da populacao:100
 Quantas geracoes deseja fazer:500
```

Figura 2 - Algoritmo Evolutivo

```
Ficheiro a ler:test
 Valor K: 4
Num Vertices: 6
Ligacoes: 7
1 - Trepa Colinas
2 - Algoritmo Evolutivo
3 - Algoritmo Hibrido
Escolha:3
 Quantas runs deseja fazer:10
 Qual o tamanho da populacao:100
 Quantas iteracoes deo trepa-colinas:1000
```

Figura 3 - Algoritmo Híbrido

Análise de Resultados

Trepa Colinas

O algoritmo trepa-colinas é um algoritmo que busca melhorar progressivamente uma solução.
Ele faz isso explorando as soluções vizinhas à atual e adotando aquelas que apresentam um desempenho (ou custo) melhor. Esse processo continua até que não sejam encontradas melhorias em um número definido de iterações, indicando que pode ter atingido uma solução de bom custo.

Trepa-colinas com Vizinhança 1

Trepa-Colinas com Vizinhança 1		100 it	1000 it	5000 it	10k it	100k it	1M it	
file1.txt	Melhor	45	45	45	45	45	45	
	MBF	89,3	59,3	60,4	57,9	55,7	56	63,1
file2.txt	Melhor	29	11	13	14	10	15	
	MBF	80,5	26	24	27,4	23,7	25,4	34,5
file3.txt	Melhor	466	408,7	336	336	336	336	
	MBF	557,1	372,6	382,5	370,4	378,2	384,5	407,55
file4.txt	Melhor	28	13	13	9	9	10	
	MBF	60,8	19,2	13,7	11,4	12,5	11,1	21,45
file5.txt	Melhor	117	20	13	14	13	14	
	MBF	154,1	35,4	24,6	23,4	18,5	19,4	45,9

Trepa-colinas com Vizinhança 2

Trepa-Colinas com Vizinhança 2		100 it	1k it	5k it	10k it	100k it	1M it	
file1.txt	Melhor	66	45	45	45	45	45	
	MBF	90,1	62,9	56,4	51,3	48,1	50,6	59,9
file2.txt	Melhor	31	18	18	15	13	15	
	MBF	84,8	33,6	24,1	24,6	17,7	16,5	33,55
file3.txt	Melhor	467	397	354	347	336	336	
	MBF	515,5	435,8	402,2	385	359,9	356,2	409,1
file4.txt	Melhor	55	29	20	13	7	7	
	MBF	97,4	41,3	26,5	18	8,4	7	33,1
file5.txt	Melhor	129	21	15	14	12	8	
	MBF	164,4	33,1	30,3	31,9	25,5	9,7	49,15

Algoritmo evolutivo

O algoritmo evolutivo é um algoritmo que busca otimizar as soluções de acordo com os princípios da evolução biológica usando processos com o crossover e a mutação para gerar novas soluções.

As soluções são avaliadas de acordo com uma função de aptidão e as mais aptas são usadas para criar uma nova geração

Evolutivo		100 Pop 100 Gen	100 Pop 500 Gen	500 Pop 100 Gen	500 Pop 500 Gen	1000 Pop 100 Gen	1000 Pop 500 Gen	
file1.txt	Melhor	45	45	45	45	45	45	
	MBF	50,2	51,9	48,1	50,3	49,1	48,6	49,7
file2.txt	Melhor	21	10	17	8	15	11	
	MBF	37,2	22,4	20,9	15,1	16,2	14,2	21
file3.txt	Melhor	345	336	336	336	336	336	
	MBF	386,6	370,4	345,2	366,7	360,1	349,3	363,05
file4.txt	Melhor	87	62	58	47	55	58	
	MBF	103.4	96.7	85,4	69,4	69,3	67,7	72,95
file5.txt	Melhor	244	384	261	247	191	146	
	MBF	514,6	584,2	549,3	406,4	321,9	251,9	438,05

Algoritmo Híbrido

O algoritmo híbrido combina ambos os algoritmos descritos acima, o trepa-colinas e o algoritmo evolutivo de modo a aproveitar as vantagens de cada um destes algoritmos.

Hibrido		100 Pop 100 it	100 Pop 500 it	100 Pop 1k it	500 Pop 100 it	500 Pop 500 it	500 Pop 1k it	
file1.txt	Melhor	45	45	45	45	45	45	
	MBF	53,2	52,9	54,3	49,3	48,8	48,4	51,15
file2.txt	Melhor	13	11	15	14	15	11	
	MBF	22,7	21,3	22,1	16,3	16,9	16,5	19,3
file3.txt	Melhor	336	336	336	336	336	336	
	MBF	364,6	369,2	361,2	366,7	339,4	340	356,85
file4.txt	Melhor	39	39	26	13	23	9	
	MBF	51,8	49,8	42,4	20,6	25,1	20,3	35
file5.txt	Melhor	151	Na	Na	Na	Na	Na	
	MBF	171,4	Na	Na	Na	Na	Na	171,4

Conclusão

Com o método Trepa-Colinas, concluímos que quanto menor o tamanho do conjunto solução, mais dificilmente atingia valores próximos do ideal e para qualquer problema necessita de milhares de iterações para obter resultados viáveis.

No método evolutivo, concluímos que para grafos com grandes quantidades de vértices e arestas é mais eficaz aumentar o tamanho da população e que produzimos melhores resultados quando o número de gerações está diretamente relacionado com o número de arestas do grafo.

Concluímos também que neste trabalho cometemos alguns erros de implementação que não fomos capazes de ultrapassar o que nos levou aos resultados obtidos.

Anexos

Spreadsheet de resultados “Resultados.xlsx”