



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Introdução à Inteligência Artificial

Licenciatura em Engenharia Informática - 2022/2023

Trabalho prático 2 – Problemas de Otimização

Docente: Patrícia Ferreira

João Miguel Baptista Duarte – a2020122715

João Miguel Duarte dos Santos – a2020136093

Coimbra, 09 de janeiro de 2022

Índice

1. Introdução	3
2. Métodos de otimização	4
2.1 Descrição dos algoritmos	4
2.1.1 Algoritmo de pesquisa local: Trepas colinas probabilístico	4
2.1.2 Algoritmo evolutivo	4
2.1.3 Algoritmo híbrido	5
3. Conclusão	5

1. Introdução

Este trabalho consiste em conceber, implementar e testar métodos de otimização que encontrem soluções de boa qualidade para diferentes instâncias do problema:

Dado um grafo e um valor inteiro k , o **Maximum Edge Subgraph Problem** consiste em encontrar um subconjunto de k -vértices tal que o número de arestas dentro do subconjunto seja máximo.

Formalmente o problema é definido:

Dados:

- um grafo não direcionado $G = (V, A)$, composto por um conjunto V de vértices ligados entre si por arestas A
- um inteiro k

Problema:

- encontrar um subconjunto de vértices S , de tamanho k , tal que $S \subseteq V$, de forma a maximizar o número de arestas desse subconjunto
- o objetivo deste problema é, portanto, de maximização.

2. Métodos de otimização

Para obter a melhor solução para um determinado grafo iremos submetê-lo a vários algoritmos de otimização.

Os métodos que iremos usar são:

- Algoritmo de pesquisa local: Trepas colinas probabilístico;
- Algoritmo evolutivo;
- Algoritmo híbrido com trepa colinas.

2.1 Descrição dos algoritmos

2.1.1 Algoritmo de pesquisa local: Trepas colinas probabilístico

Muito em semelhança ao trepa colinas “first choice”, este algoritmo destaca-se por definir uma probabilidade no início do programa, que após calcular o custo do vizinho gera um número aleatório entre 1 e o número de vértices, caso o número obtido seja menor que a probabilidade definida no início, substitui a solução obtida e iguala o custo ao custo do vizinho.

Esta probabilidade definida, apesar de estática, é uma estratégia de reparação, o que permite obter resultados um pouco melhores que o “first choice”.

2.1.2 Algoritmo evolutivo

Este algoritmo aplica o princípio de sobrevivência do indivíduo mais apto, de uma determinada população, de forma a gerar melhores soluções.

Em cada geração (GER) de um GA é criado um novo conjunto de soluções pelo processo de seleção de indivíduos de acordo com seu nível de aptidão. Estes indivíduos são posteriormente reproduzidos, utilizando operadores importados de mecanismos de adaptação natural, tais como, a mutação e o crossover.

2.1.3 Algoritmo híbrido

Este algoritmo aplica o método evolutivo e ao mesmo tempo executa um algoritmo de pesquisa local (trepa colinas probabilístico) para refinar a solução final obtida.

3. Conclusão

Trepa Colinas com vizinhança 1 sem aceitar solucoes com igual custo

NOME FICH		100 i	1000 i	5000 i	10000 i
file1	Melhor	20	20	20	20
	MBF	20.0000	20.0000	20.0000	20.0000
file2	Melhor	15	15	15	15
	MBF	14.880000	14.871000	14.869000	14.796400
file3	Melhor	112	112	112	112
	MBF	110.150002	110.484001	110.558403	110.547203
file4	Melhor	79	79	79	79
	MBF	76.390000	75.723000	75.837200	75.923401
file5	Melhor	98	98	98	98
	MBF	89.930000	89.596001	89.677399	89.551903

Trepa Colinas com vizinhança 2 sem aceitar solucoes com igual custo

NOME FICH		100 i	1000 i	5000 i	10000 i
file1	Melhor	20	20	20	20
	MBF	20.0000	20.0000	20.0000	20.0000
file2	Melhor	15	15	15	15
	MBF	15.0000	14.997000	15.0000	15.0000
file3	Melhor	112	112	112	112
	MBF	110.800000	110.635000	110.690800	110.718200
file4	Melhor	79	79	79	79
	MBF	68.890000	68.156000	68.248000	68.234900
file5	Melhor	87	93	98	98
	MBF	75.080000	75.525000	75.729000	75.998000

Trepa Colinas Probabilístico com vizinhança 1 e aceitando solucoes com igual custo

NOME FICH		100 i	1000 i	5000 i	10000 i
file1	Melhor	20	20	20	20
	MBF	19.340000	19.495001	19.525801	19.522301
file2	Melhor	15	15	15	15
	MBF	14.270000	14.326000	14.324800	14.35900
file3	Melhor	112	112	112	112
	MBF	108.059998	108.116997	108.067001	108.123400
file4	Melhor	79	79	79	79
	MBF	54.330002	53.813999	54.601002	55.402400
file5	Melhor	76	90	93	97
	MBF	47.900002	49.854000	49.101601	51.532500

Trepa Colinas Probabilístico com vizinhança 2 e aceitando solucoes com igual custo

NOME FICH		100 i	1000 i	5000 i	10000 i
file1	Melhor	20	20	20	20
	MBF	19.240000	19.086000	19.091600	19.090200
file2	Melhor	15	15	15	15
	MBF	13.170000	13.211000	14.330000	14.543000
file3	Melhor	109	110	112	112
	MBF	104.089996	103.943001	104.044197	105.239201
file4	Melhor	62	79	72	79
	MBF	39.880001	54.595001	38.018799	40.502430
file5	Melhor	90	85	93	96
	MBF	47.840000	48.992001	49.216400	51.532700

De um modo geral, podemos concluir que nos aproximamos do melhor custo consoante aumentamos o número de iterações.