



Instituto Superior de Engenharia

Politécnico de Coimbra

**Departamento de Engenharia Informática e de
Sistemas**

Problema de Otimização

Chelsea Carina Pais da Cunha Duarte - 2021100010

Lucas Ribeiro Caetano – 2020132564

Coimbra

2022



**Instituto Superior
de Engenharia**

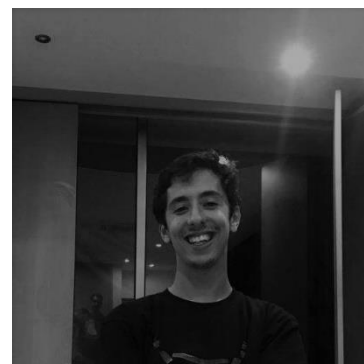
Politécnico de Coimbra

**Licenciatura em Engenharia
Informática
Introdução à Inteligência
Artificial**

**RELATÓRIO DO
TRABALHO PRÁTICO DE
PROBLEMAS DE
OTIMIZAÇÃO -
“Maximum Edge Subgraph
Problem”**



Chelsea Carina Pais
da Cunha Duarte



Lucas Ribeiro Caetano

Índice

1. INTRODUÇÃO	4
2. APRESENTAÇÃO DOS DIVERSOS ALGORITMOS	3
2.1. 1º ALGORITMO: TREPA COLINAS	6
2.1.1. O que é o algoritmo Trepas Colinas e como funciona	6
2.1.2. Quais são as vantagens e desvantagens do algoritmo Trepas Colinas	6
2.2. 2º ALGORITMO: MÉTODO EVOLUTIVO	7
2.2.1. O que é o método Evolutivo e como funciona	7
2.2.2. Quais são as vantagens e desvantagens do método Evolutivo	7
2.3. 3º ALGORITMO: MÉTODO HÍBRIDO.....	8
2.2.1. O que é o método Híbrido e como funciona.....	8
2.2.2. Quais são as vantagens e desvantagens do método Híbrido.....	8
3.1. ANÁLISE DE RESULTADOS E ESTUDO DE COMPARAÇÃO	9
3.1.1. Resultados do 1º Algoritmo.....	10
3.1.2. Resultados do 2º Algoritmo.....	11
3.1.2. Resultados do 3º Algoritmo.....	12
3.1.3 Estudo de Comparação	13
4 CONCLUSÃO.....	14

1 INTRODUÇÃO

Inteligência artificial (IA) é a área da tecnologia que se dedica ao estudo e ao desenvolvimento de sistemas capazes de realizar tarefas que exigem inteligência humana. Isso inclui tarefas como raciocínio, percepção, aprendizagem e adaptação.

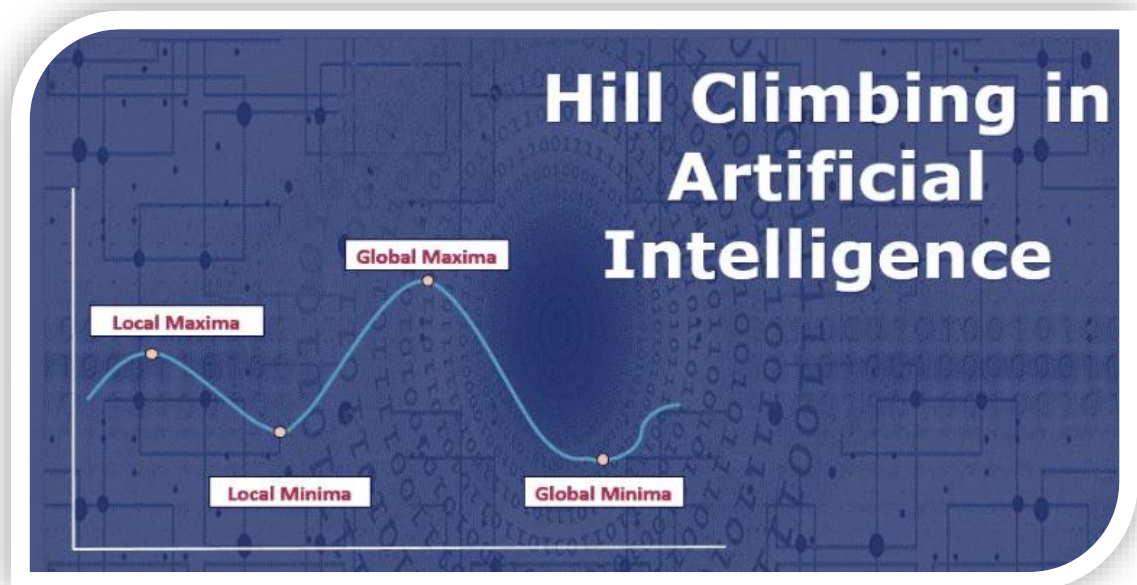
Ao longo do nosso trabalho, desenvolvemos três algoritmos diferentes: trepa colinas, evolutivo e híbrido. O algoritmo trepa colinas é um método utilizado para encontrar o mínimo ou o máximo de uma função num determinado campo de valores. O algoritmo evolutivo é um método de otimização baseado em princípios de seleção natural, onde uma população de soluções é gerada aleatoriamente e as soluções mais aptas são selecionadas para sobreviver e gerar novas gerações. Por fim, o algoritmo híbrido, no nosso caso, é uma combinação dos dois algoritmos referidos anteriormente, com o objetivo de aproveitar as vantagens de cada um deles.

Neste relatório, será abordada a diferença detalhada entre todos os métodos, analisando os resultados obtidos. Para além disso, iremos apresentar o modo de funcionamento de cada algoritmo e as aplicações práticas dos mesmos, assim como as suas possíveis limitações.

CAPÍTULO 2

APRESENTAÇÃO DOS DIVERSOS ALGORITMOS

2.1. TREPA COLINAS



2.1.1. O que é o algoritmo Trepa Colinas e como funciona

O algoritmo trepa colinas é um método de otimização utilizado em diversas áreas da computação e da ciência. O seu nome é devido ao facto de o objetivo ser encontrar o ponto mais alto de uma "colina" de valores.

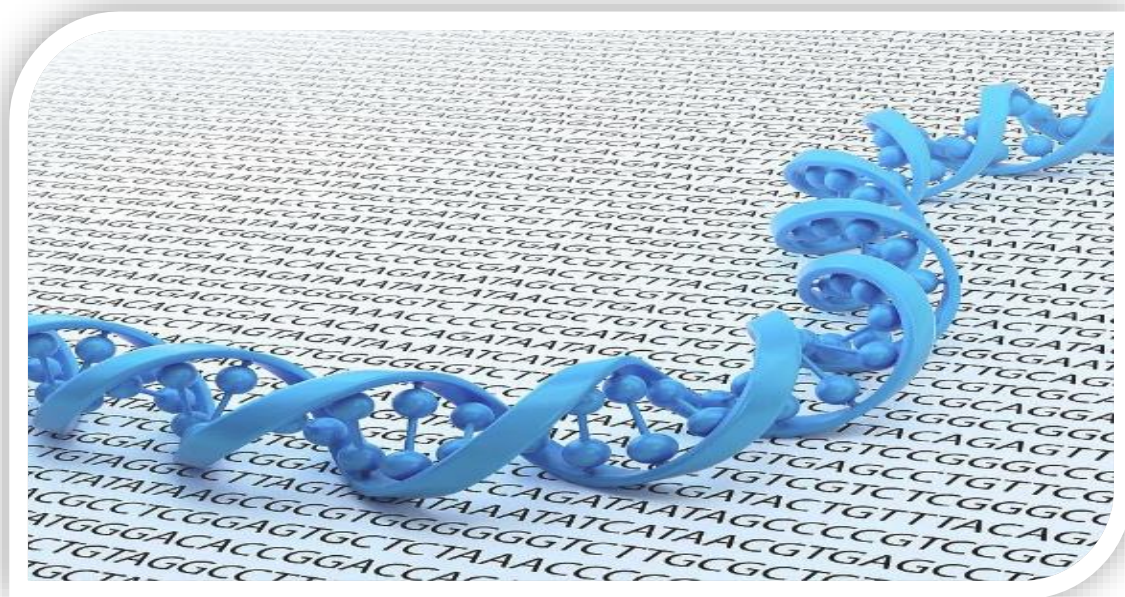
Este algoritmo assenta numa premissa simples: O objetivo é chegar ao ponto mais alto, e para isso, é imperativo avaliar as possibilidades vizinhas. É assim que o algoritmo trepa colinas funciona: ele começa em um ponto aleatório e, a cada iteração, vai para a direção em que a altitude é superior ao ponto atual, até chegar ao topo da colina. Trata-se de um algoritmo bastante simples de implementar, porém assume-se ineficiente perante problemas de complexidade superior.

2.1.1.1. Quais são as vantagens e desvantagens do algoritmo Trepa Colinas

Vantagens: Simplicidade: o algoritmo é relativamente simples de implementar e entender; Eficiência: o algoritmo é geralmente rápido em encontrar o mínimo ou o máximo de uma função; Robustez: o algoritmo é menos propenso a ficar preso em mínimos locais do que outros algoritmos, como o gradiente descendente.

Desvantagens: Sensibilidade às condições iniciais, o algoritmo pode ter resultados diferentes dependendo de onde é iniciado; Dificuldade em lidar com funções não convexas, o algoritmo só pode encontrar o mínimo ou o máximo global de uma função convexa. Não é garantido que o algoritmo encontrará a solução ótima, o algoritmo pode parar em um mínimo ou máximo local, em vez do global.

2.1. MÉTODO EVOLUTIVO



2.1.1. O que é o Método Evolutivo e como funciona

O método evolutivo é um tipo de algoritmo de otimização que se baseia em princípios da evolução natural, como a seleção e a reprodução. Ele é usado para encontrar a solução ótima para um determinado problema por meio de várias gerações de soluções.

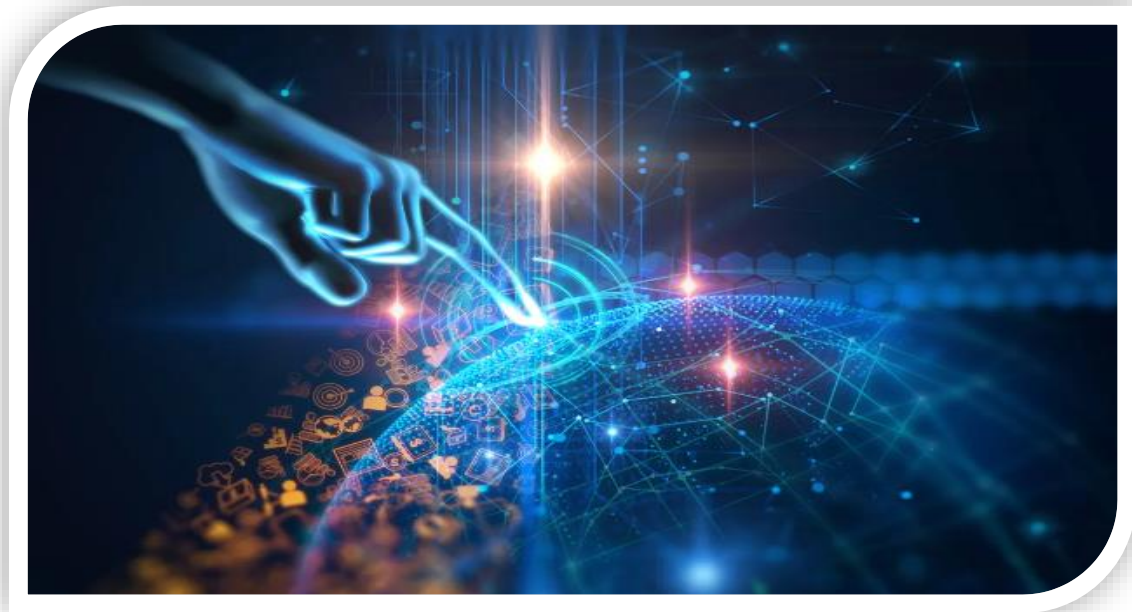
O método evolutivo é geralmente utilizado em problemas em que é difícil encontrar a solução ótima de forma exata, como problemas de otimização não lineares ou problemas com muitas variáveis. Os métodos de Seleção utilizados foram os de Torneio e Fitness. Os métodos de Mutação foram Mutação Binária e Mutação por Troca. Os métodos de recombinação foram o recombinação com 2 pontos de recorte e recombinação uniforme.

2.1.1.1. Quais são as vantagens e desvantagens do Método Evolutivo

Vantagens: As vantagens do Método Evolutivo são idênticas às do algoritmo anterior. Para além das vantagens referidas anteriormente, este algoritmo oferece melhores soluções em problemas mais complexos.

Desvantagens: Sensibilidade às condições iniciais, o algoritmo pode ter resultados diferentes dependendo de onde é iniciado; Dificuldade em lidar com funções não convexas, o algoritmo só pode encontrar o mínimo ou o máximo global de uma função convexa. Não é garantido que o algoritmo encontrará a solução ótima, o algoritmo pode parar em um mínimo ou máximo local, em vez do global.

2.1. MÉTODO HÍBRIDO



2.1.1. O que é o Método Híbrido e como funciona

O Método Híbrido é uma técnica de otimização que combina dois ou mais algoritmos para resolver um problema. Um exemplo de método híbrido é a combinação do Algoritmo Trepas Colinas com o Algoritmo Evolutivo.

O Método Híbrido combina esses dois algoritmos de forma a aproveitar os pontos fortes de cada um. O Algoritmo Trepas Colinas é rápido e eficiente em encontrar soluções próximas da ótima, enquanto o Algoritmo Evolutivo é capaz de explorar mais amplamente o espaço de busca e encontrar soluções mais distantes da solução inicial.

2.1.1.1. Quais são as vantagens e desvantagens do Método Híbrido

Vantagens: Maior eficiência, como o Método Híbrido combina dois ou mais algoritmos, ele pode aproveitar os pontos fortes de cada um e, assim, ser mais eficiente em encontrar uma solução ótima; Boa performance em problemas de grande escala, este método pode ser eficiente em problemas com um grande número de variáveis, pois permite a combinação de algoritmos que exploram de forma eficiente o espaço de busca.

Desvantagens: Complexidade, o Método Híbrido é mais complexo do que algoritmos de otimização simples, pois envolve a combinação de dois ou mais algoritmos. Isso pode dificultar a implementação e o entendimento do método; Dificuldade de interpretação dos resultados: Como o Método Híbrido combina dois ou mais algoritmos, os resultados podem ser difíceis de interpretar e pode ser difícil entender como cada algoritmo contribuiu para o resultado final.

CAPÍTULO 3

ANÁLISE DE RESULTADOS E ESTUDO DE COMPARAÇÃO

3.1. RESULTADOS DO 1º ALGORITMO: TREPA COLINAS

Trepacolas com vizinhança 1 e com 15 rondas					
Ficheiro		50 iterações	100 iterações	250 iterações	500 iterações
teste.txt	Melhor	5	5	5	5
	MBF	4.900000	4.94000	5.0000	5.0000
file1.txt	Melhor	20	20	20	20
	MBF	19.37001	19.770000	19.990000	20.0000
file2.txt	Melhor	15	15	15	15
	MBF	13.250000	14.070000	14.720000	14.740000
file3.txt	Melhor	107	109	112	115
	MBF	103.129997	105.190003	106.94098	108.339996
file4.txt	Melhor	40	45	72	79
	MBF	22.30001	29.940001	47.150002	61.810001
file5.txt	Melhor	56	82	89	94
	MBF	45.531250	66.531250	72.7203125	91.062500

Trepacolas com vizinhança 2 e aceitando soluções de custo igual - 30 rondas					
Ficheiro		50 iterações	100 iterações	250 iterações	500 iterações
teste.txt	Melhor	5	5	5	5
	MBF	4.980000	5.000000	5.0000	5.0000
file1.txt	Melhor	20	20	20	20
	MBF	19.070000	19.340000	19.610001	19.830000
file2.txt	Melhor	15	15	15	15
	MBF	12.530000	13.440000	14.380000	14.820000
file3.txt	Melhor	105	107	109	109
	MBF	101.300003	102.889999	104.919998	105.830002
file4.txt	Melhor	36	45	64	79
	MBF	18.10000	24.640001	31.18000	68.430001
file5.txt	Melhor	44	46	58	86
	MBF	20.531250	23.50000	31.02000	39.779999

Com base nas tabelas acima, pudemos concluir que o maior valor foi registado foi 115 utilizando a Vizinhança 1 com 15 Rondas. Para além disso, conseguimos constatar que perante o aumento de iterações se dá um melhoramento da solução final, assim como uma melhor média.

Ficheiro teste.txt		Algoritmo base sem penalização										Algoritmo base com penalização e reparação									
Parâmetros		Operadores de reconhecimento					Operadores de mutação					Operadores de reconhecimento					Operadores de mutação				
		Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas	Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas	Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas					
Reconhecimento	Reparação e correção	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo				
reconhecimento - 1000	penal. considerando - 0.0	1	5.000	5	5.000	5	0	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000				
reconhecimento - 10	penal. considerando - 0.0	1	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000				
penal. unico - 0.0	penal. considerando - 0.0	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000				
penal. unico - 0.0	penal. unico - 0.0	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000				
reconhecimento - 100	penal. unico - 0.0, penal. e penalizacao - 0	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000				
penal. considerando - 0.0	penal. unico - 0.0	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000	5	5.000				
Variação de teste																					
Ficheiro file.txt		Algoritmo base sem penalização										Algoritmo base com penalização e reparação									
Parâmetros		Operadores de reconhecimento					Operadores de mutação					Operadores de reconhecimento					Operadores de mutação				
		Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas	Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas	Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas					
Reconhecimento	Reparação e correção	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo				
reconhecimento - 1000	penal. considerando - 0.0	10	10.000	10	10.000	10	10.0000	10	10.000	10	10.000	10	10.000	10	10.000	10	10.000				
reconhecimento - 10	penal. considerando - 0.0	10	10.0000	10	10.000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
penal. unico - 0.0	penal. considerando - 0.0	10	10.0000	10	10.000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
penal. unico - 0.0	penal. considerando - 0.0	10	10.0000	10	10.000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
reconhecimento - 100	penal. unico - 0.0, penal. e penalizacao - 0	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
penal. considerando - 0.0	penal. unico - 0.0	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
Variação de teste																					
Ficheiro file.txt		Algoritmo base sem penalização										Algoritmo base com penalização e reparação									
Parâmetros		Operadores de reconhecimento					Operadores de mutação					Operadores de reconhecimento					Operadores de mutação				
		Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas	Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas	Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas					
Reconhecimento	Reparação e correção	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo				
reconhecimento - 1000	penal. considerando - 0.0	10	10.0000	10	10.0000	10	10.00000	10	10.0000	10	10.00	10	10.0000	10	10.0000	10	10.0000				
reconhecimento - 10	penal. considerando - 0.0	10	10.0000	10	10.0000	10	10.00000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
penal. unico - 0.0	penal. considerando - 0.0	10	10.0000	10	10.0000	10	10.00000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
penal. unico - 0.0	penal. considerando - 0.0	10	10.0000	10	10.0000	10	10.00000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
reconhecimento - 100	penal. unico - 0.0, penal. e penalizacao - 0	10	10.00000	10	10.0000	10	10.00000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
penal. considerando - 0.0	penal. unico - 0.0	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000	10	10.0000				
Variação de teste																					
Ficheiro file.txt		Algoritmo base sem penalização										Algoritmo base com penalização e reparação									
Parâmetros		Operadores de reconhecimento					Operadores de mutação					Operadores de reconhecimento					Operadores de mutação				
		Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas	Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas	Grupos	Grupos e 2 parâmetros	Grupos sem avaliação	Método Bruto	Método Por Tarefas					
Reconhecimento	Reparação e correção	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo	Metodo				
reconhecimento - 1000	penal. considerando - 0.0	100	100.0000	100	100.00000	100	100.00000	100	100.000	100	100.0000	100	100.0000	100	100.0000	100	100.00000				
reconhecimento - 10	penal. considerando - 0.0	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000				
penal. unico - 0.0	penal. considerando - 0.0	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000				
penal. unico - 0.0	penal. considerando - 0.0	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000				
reconhecimento - 100	penal. unico - 0.0, penal. e penalizacao - 0	100	100.00000	100	100.0000	100	100.00000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000				
penal. considerando - 0.0	penal. unico - 0.0	100	100.00000	100	100.0000	100	100.00000	100	100.0000	100	100.0000	100	100.0000	100	100.0000	100	100.0000				
Variação de teste																					

Com base na análise dos dados, averiguamos que no ficheiro teste, os valores permaneceram iguais devido ao facto de quanto menor o número de vértices maior é a exatidão do algoritmo. Se tivermos os parâmetros fixos e aumentarmos o valor do número de participantes do torneio, o algoritmo chega aos melhores resultados. Relativamente à probabilidade de mutação na maior parte dos casos se esse valor for aumentado o melhor valor encontrado é geralmente igual ou superior ao valor atual.

3.4. ESTUDO DE COMPARAÇÃO



Consoante a análise dos resultados, podemos constatar que o método Evolutivo é o método mais eficiente a alcançar a melhor solução.

4 CONCLUSÃO

Concluindo, durante a realização deste trabalho, ambos incorporamos os fundamentais da disciplina assim como toda um reforço na aprendizagem da linguagem C. Testamos vários ficheiros de texto, dado o problema do enunciado e conseguimos atribuir diversas soluções consoante o método utilizado. Sendo Inteligência Artificial uma área relativamente recente, podemos adquirir conhecimentos acerca do seu meio, assim como alguns algoritmos usados.