



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra

Licenciatura em Engenharia Informática

RELATÓRIO Do Trabalho Prático nº 2 -  
Problema de Otimização

## **Coin Change Problem**

Autores

**Duarte Xavier de Oliveira Santos 2022149622**

**Gustavo Trigo Costa 2023145800**



INSTITUTO POLITÉCNICO  
DE COIMBRA

INSTITUTO SUPERIOR  
DE ENGENHARIA  
DE COIMBRA

Coimbra, dezembro de 2024



## ÍNDICE

1.	
1.Introdução .....	2
2. Representação do Problema, Função de Avaliação e Objetivo.....	2
3. Descrição dos Algoritmos e Heurísticas .....	3
4. Experimentação e Análise de Resultados.....	4
5. Conclusão .....	9

# 1.INTRODUÇÃO

Este relatório documenta o desenvolvimento e os resultados do Trabalho Prático 2, cujo objetivo foi resolver o problema da combinação ótima de moedas para atingir um determinado valor (solução ótima), utilizando diferentes abordagens de otimização. Foram exploradas técnicas de heurística (trepa-colinas), algoritmos evolutivos (mutação, recombinação e seleção) e abordagens híbridas que combinam os dois métodos.

## 2. REPRESENTAÇÃO DO PROBLEMA, FUNÇÃO DE AVALIAÇÃO E OBJETIVO

### Representação das Soluções

As soluções foram representadas como vetores inteiros de tamanho  $n$ , onde cada posição do vetor representa o número de vezes que uma determinada moeda é usada. Por exemplo, para um vetor  $[2, 1, 0]$  e um conjunto de moedas  $\{1\text{€}, 2\text{€}, 5\text{€}\}$ , a solução indica que são usadas duas moedas de 1€ e uma moeda de 2€, totalizando um valor de 4€, sendo o custo assim a soma da quantidade de moedas, neste caso 3.

### Função de Avaliação

A função de avaliação mede a qualidade de uma solução com base em dois critérios principais:

**Validação:** Verifica se a solução atinge o valor objetivo  $V$ . Soluções inválidas (que não atingem  $V$ ) recebem uma penalização alta.

**Custo:** Para soluções válidas, é calculado o custo total como a soma do número de moedas usadas. O objetivo é minimizar este custo.

A função de fitness é formulada da seguinte forma:

Para soluções válidas:  $\text{Fitness} = \text{Custo}$

Para soluções inválidas:  $\text{Fitness} = \text{Fitness} + \text{Penalização}$ , sendo a penalização calculada da seguinte forma:

$\text{Penalização} = 1000 + \text{fabs}(\text{somaV} - V) * 100$  (penalização não cega);

$\text{Penalização} = 1000$  (penalização cega);

### 3. DESCRIÇÃO DOS ALGORITMOS E HEURÍSTICAS

#### Trepa-Colinas

- **Solução Inicial:** Gera-se uma solução aleatória (entre 0 a 10) onde as moedas são distribuídas sem nenhuma validação inicial de viabilidade.
- **Vizinhança:** É definida pela alteração de uma unidade entre duas posições diferentes do vetor de solução. Por exemplo, retirar uma moeda de uma posição e adicioná-la a outra.
- **Critério de Avaliação:** Uma nova solução é aceita se tiver um custo menor que a atual.
- **Iterações:** O algoritmo é executado por um número fixo de iterações .  
O trepa-colinas também foi utilizado em abordagens híbridas, descritas abaixo.

#### Algoritmo Evolutivo

O algoritmo evolutivo foi configurado com os seguintes componentes:

- **Representação e Parâmetros**
- **Indivíduos:** Cada indivíduo da população representa uma solução no formato descrito acima (vetores inteiros).
- **Parâmetros:** Tamanho da população = 100, número de gerações = 2500, probabilidade de recombinação = 0.7, probabilidade de mutação = 0.001, tamanho do torneio = 2.
- **Operadores Genéticos**
- **Seleção:** Foi implementado o torneio simples e o torneio com múltiplos elementos para selecionar os pais.
- **Crossover:** Implementação de crossover de 1 ponto e crossover uniforme.
- **Mutação:** Mutação simples (alteração de uma posição aleatória) e por troca (troca de valores entre duas posições diferentes).
- **Reparo das Soluções**

- Uma função de reparo é aplicada para corrigir indivíduos inválidos que não atingem o valor  $V$ , ajustando as moedas de forma a alcançar o valor objetivo, isto é caso a soma total do valor das moedas for inferior ao  $V$ , ele irá adicionar moedas. Senão, irá retirar moedas.

- **Abordagens Híbridas**

Foram testadas duas abordagens híbridas que combinam o algoritmo evolutivo com o trepa-colinas:

**Abordagem 1:** Aplica o trepa-colinas a cada indivíduo da população inicial antes de iniciar o processo evolutivo.

**Abordagem 2:** Aplica o trepa-colinas ao melhor indivíduo de cada geração.

As duas abordagens foram comparadas em termos de desempenho.

**Justificação das Escolhas**

- O trepa-colinas foi escolhido por ser eficiente para ajustar soluções em um espaço de pesquisa mais limitado.
- O algoritmo evolutivo foi usado para explorar um espaço de pesquisa maior e evitar o risco de ficar preso em soluções que parecem boas, mas que não são ótimas..
- A combinação dos dois métodos procura tirar partido do trepa-colinas para ajustes locais e do evolutivo para assegurar uma pesquisa mais global e diversificada.

## 4. EXPERIMENTAÇÃO E ANÁLISE DE RESULTADOS

Neste tópico, será analisado o desempenho de todas as abordagens implementadas, incluindo o trepa-colinas, o algoritmo evolutivo e as abordagens híbridas. No caso do trepa-colinas, serão testadas variações como diferentes definições de vizinhança (vizinhança 1 e 2) e critérios de aceitação de soluções (aceitando ou rejeitando soluções com custo igual), avaliando o impacto dessas alterações na eficiência do método. Para o algoritmo evolutivo, será examinada a influência de diferentes

combinações de operadores (como penalização cega, reparação, e tipos de recombinação e mutação), bem como a variação de parâmetros como probabilidade de recombinação (pr), probabilidade de mutação (pm), tamanho da população e número de gerações. Por fim, serão avaliadas as abordagens híbridas, comparando os dois métodos testados, variando a probabilidade de gerar vizinhos. Estas análises procuram identificar as configurações mais eficazes para cada algoritmo.

**TABELA 1 - Trepa-Colinas com Vizinhança 1 (RUNS 20)**

Nome Ficheiro		100 it	1000 it	5000 it	10000 it
file1.txt	Melhor	2	2	2	2
	MBF	4.40	4.70	3.80	5.30
file2.txt	Melhor	32	31	41	32
	MBF	50.35	50.90	53.35	48.30
file3.txt	Melhor	173	177	172	172
	MBF	189.65	186.20	188.95	186.60
file4.txt	Melhor	350	344	324	361
	MBF	395.55	392.70	388.50	394.55
file5.txt	Melhor	4	4	4	4
	MBF	5.55	7.25	6.95	6.30

**TABELA 2 - Trepa-Colinas com Vizinhança 1 e Aceitando Soluções de Custo Igual (RUNS 20)**

Nome Ficheiro		100 it	1000 it	5000 it	10000 it
file1.txt	Melhor	2	2	2	2
	MBF	5.90	4.10	5.45	5.30
file2.txt	Melhor	37	34	36	38
	MBF	51.00	48.90	47.45	48.55
file3.txt	Melhor	173	165	162	175
	MBF	187.05	184.15	183.70	187.70
file4.txt	Melhor	343	329	353	353
	MBF	392.95	389.85	390.90	401.95
file5.txt	Melhor	4	4	4	4
	MBF	5.90	6.60	5.80	5.05

**TABELA 3 - Trepa-Colinas com Vizinhaça 2 (RUNS 20)**

Nome Ficheiro		100 it	1000 it	5000 it	10000 it
file1.txt	Melhor	2	2	2	2
	MBF	6.50	5.30	4.40	5.60
file2.txt	Melhor	34	28	34	34
	MBF	46.10	48.00	47.55	46.70
file3.txt	Melhor	157	174	175	162
	MBF	187.65	189.05	190.50	187.15
file4.txt	Melhor	347	350	331	315
	MBF	395.00	393.15	386.35	393.85
file5.txt	Melhor	4	4	4	4
	MBF	7.40	6.05	5.50	6.35

**TABELA 4 - Trepa-Colinas com Vizinhaça 2 e Aceitando Soluções de Custo Igual (RUNS 20)**

Nome Ficheiro		100 it	1000 it	5000 it	10000 it
file1.txt	Melhor	2	2	2	2
	MBF	5.15	5.30	6.65	5.00
file2.txt	Melhor	27	35	34	38
	MBF	45.55	51.50	46.75	47.95
file3.txt	Melhor	164	169	169	154
	MBF	187.00	186.20	191.55	193.35
file4.txt	Melhor	345	351	345	343
	MBF	395.05	389.65	387.70	386.65
file5.txt	Melhor	4	4	4	4
	MBF	5.70	6.10	5.50	6.55

## Conclusão Geral sobre o Trepa-Colinas

O algoritmo Trepa-Colinas, na versão com vizinhaça 1 ou 2, mostra bom desempenho em instâncias simples, onde é capaz de alcançar a solução ótima de forma consistente. No entanto, em instâncias mais complexas, ele apresenta variações significativas nos resultados e dificuldades em encontrar soluções ótimas ou próximas da ótima, refletidas nas flutuações no valor da função de avaliação (MBF). O impacto das variações nas vizinhanças e da aceitação de soluções de custo igual parece ser mais perceptível em problemas de maior complexidade, mas não gera uma melhoria substancial nas soluções encontradas.



**TABELA 5 - Algoritmo Evolutivo com Variações de Operadores (Recombinação, Mutação, Penalização, Reparação) e Parâmetros (Probabilidade de Recombinação, Probabilidade de Mutação, Tamanho da População) no file1.txt**

Ficheiro file1.txt											
Parâmetros Fixos	Parâmetros a variar	Algoritmo base (Recombinação de 1 ponto de corte + Mutação + Penalização cega)		Recombinação de 1 ponto de corte + Mutação + Penalização não cega		Recombinação de 1 ponto de corte + Mutação + Reparação		Recombinação de 1 ponto de corte + Mutação por troca + Reparação		com recombinação uniforme + mutação + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (ger = 2500)	pr = 0.3	2,0	2,0	2,0	3.5	2,0	2,0	2,0	2,0	2,0	2,0
pm = 0.01	pr = 0.5	2,0	2,0	2,0	3.5	2,0	2,0	2,0	2,0	2,0	2,0
tsize = 2	pr = 0.7	2,0	2,0	2,0	3.5	2,0	2,0	2,0	2,0	2,0	2,0
pop = 100 (ger = 2500)	pm = 0.0	8,0	9,5	8,0	643,0	2,0	2,0	2,0	2,0	2,0	2,0
pop = 100	pm = 0.001	2,0	6,5	5,0	5,0	2,0	2,0	2,0	2,0	2,0	2,0
pr = 0.7	pm = 0.01	2,0	3,5	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0
tsize = 2	pm = 0.05	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0
pr = 0.7	pop = 10 (ger = 7.5K)	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0
pm = 0.05	pop = 50 (ger = 5K)	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0
tsize = 2	pop = 100 (ger = 2.5K)	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0
Parâmetros Fixos	Parâmetros a variar	Algoritmo base		com penalização		com reparação		com mutação por troca + reparação		com recombinação uniforme + mutação binária + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (gen = 2500)	tsize = 3	5,0	5,0	5,0	6,5	2,0	2,0	2,0	2,0	2,0	2,0
pr = 0.7	tsize = 10	2,0	5,0	8,0	8,0	2,0	2,0	2,0	2,0	2,0	2,0
pm = 0.001	tsize = 50	2,0	5,0	5,0	5,0	2,0	2,0	2,0	2,0	2,0	2,0

**TABELA 6 - Algoritmo Evolutivo com Variações de Operadores (Recombinação, Mutação, Penalização, Reparação) e Parâmetros (Probabilidade de Recombinação, Probabilidade de Mutação, Tamanho da População) no file2.txt**

Ficheiro file2.txt											
Parâmetros Fixos	Parâmetros a variar	Algoritmo base (Recombinação de 1 ponto de corte + Mutação + Penalização cega)		Recombinação de 1 ponto de corte + Mutação + Penalização não cega		Recombinação de 1 ponto de corte + Mutação + Reparação		Recombinação de 1 ponto de corte + Mutação por troca + Reparação		com recombinação uniforme + mutação + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (ger = 2500)	pr = 0.3	1014,0	1018,5	1355,0	1554,0	20,0	21,0	20,0	20,0	20,0	20,0
pm = 0.01	pr = 0.5	1015,0	1020,0	1605,00	1647,00	20,0	20,0	20,0	20,0	20,0	20,0
tsize = 2	pr = 0.7	1015,0	1015,5	1305,00	1438,00	20,0	20,0	20,0	20,0	20,0	20,0
pop = 100 (ger = 2500)	pm = 0.0	1016,0	1020,0	1389,0	1500,5	20,0	20,0	20,0	24,5	27,0	28,0
pop = 100	pm = 0.001	1013,0	1020,5	1532,00	1586,50	20,0	20,0	20,0	20,0	20,0	20,0
pr = 0.7	pm = 0.01	1012,0	1015,0	1475,00	1686,0	20,0	20,0	20,0	20,0	20,0	21,0
tsize = 2	pm = 0.05	1018,0	1019,0	1738,00	1762,0	20,0	20,0	20,0	20,0	20,0	20,0
pr = 0.7	pop = 10 (ger = 7.5K)	1026,0	1026,5	1335,0	1401,0	20,0	20,0	20,0	20,0	20,0	21,0
pm = 0.05	pop = 50 (ger = 5K)	1015,0	1019,00	1715,0	1724,0	20,0	20,0	20,0	20,0	20,0	21,0
tsize = 2	pop = 100 (ger = 2.5K)	1024,0	1025,0	1422,0	1548,5	20,0	20,0	20,0	20,0	23,0	23,0
Parâmetros Fixos	Parâmetros a variar	Algoritmo base		com penalização		com reparação		com mutação por troca + reparação		com recombinação uniforme + mutação binária + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (gen = 2500)	tsize = 3	1018,0	1019,0	1508,0	1663,0	20,0	20,0	20,0	20,0	20,0	20,0
pr = 0.7	tsize = 10	1013,0	1019,0	1592,0	1689,0	20,0	20,0	20,0	20,0	20,0	20,0
pm = 0.001	tsize = 50	1023,0	1023,5	1843,0	1855,0	20,0	20,0	20,0	20,0	20,0	20,0

**TABELA 7 - Algoritmo Evolutivo com Variações de Operadores (Recombinação, Mutação, Penalização, Reparação) e Parâmetros (Probabilidade de Recombinação, Probabilidade de Mutação, Tamanho da População) no file3.txt**

Ficheiro file3.txt											
Parâmetros Fixos	Parâmetros a variar	Algoritmo base (Recombinação de 1 ponto de corte + Mutação + Penalização cega)		Recombinação de 1 ponto de corte + Mutação + Penalização não cega		Recombinação de 1 ponto de corte + Mutação + Reparação		Recombinação de 1 ponto de corte + Mutação por troca + Reparação		com recombinação uniforme + mutação + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (ger = 2500) pm = 0.01 tsize = 2	pr = 0.3	1018,0	1023,5	10300.00	10355.00	103,0	103.5	102,0	102,0	102,0	102,0
	pr = 0.5	1009,0	1015,0	9785.00	9917.00	102,0	103,0	102,0	102,0	103,0	103,5
	pr = 0.7	1013,0	1017,5	10103.00	10148.500	103,0	103,0	102,0	102,0	102,0	102,0
pop = 100 (ger = 2500) pm = 100 pr = 0.7 tsize = 2	pm = 0.0	1014,0	1014,5	10007.0	10007.00	127.00	133.00	135,0	138,0	161,0	163,0
	pm = 0.001	1009,0	1014,5	9954.00	10188.50	102,0	102,0	106,0	109,0	103,0	103,5
	pm = 0.01	1004,0	1012,0	10503.00	10630.0	102,0	102,5	102,0	102,0	102,0	102,5
pr = 0.7 pm = 0.05 tsize = 2	pm = 0.05	1013,0	1014,0	9763.00	10353.50	107,0	107,0	102,0	102,0	106,0	115,0
	pop = 10 (ger = 7.5K)	1018.00	1019.50	9692.0	9771.0	102,0	102,0	102,0	102,0	102,0	102,0
	pop = 50 (ger = 5K)	1016,0	1020.00	9705.00	10011.0	102,0	102,0	102,0	102,0	107,0	108,0
pm = 0.05 tsize = 2	pop = 100 (ger = 2.5K)	1017,0	1018.5	10351.0	10406.0	102,0	103,0	102,0	102,0	107,0	109.5
Parâmetros Fixos	Parâmetros a variar	Algoritmo base		com penalização		com reparação		com mutação por troca + reparação		com recombinação uniforme + mutação binária + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (gen = 25) pr = 0.7 pm = 0.001	tsize = 3	1020,0	1020,5	9813.00	9985.5	102,0	102,0	102,0	112,0	103,0	103,5
	tsize = 10	1008,0	1014,0	9782.0	10005.5	102,0	102,0	102,0	119,0	104,0	104,5
	tsize = 50	1001,0	1001,0	10567.0	10633.5	102,0	102,0	126,0	145,0	102,0	103,5

**TABELA 8 - Algoritmo Evolutivo com Variações de Operadores (Recombinação, Mutação, Penalização, Reparação) e Parâmetros (Probabilidade de Recombinação, Probabilidade de Mutação, Tamanho da População) no file4.txt**

Ficheiro file4.txt											
Parâmetros Fixos	Parâmetros a variar	Algoritmo base (Recombinação de 1 ponto de corte + Mutação + Penalização cega)		Recombinação de 1 ponto de corte + Mutação + Penalização não cega		Recombinação de 1 ponto de corte + Mutação + Reparação		Recombinação de 1 ponto de corte + Mutação por troca + Reparação		com recombinação uniforme + mutação + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (ger = 2500) pm = 0.01 tsize = 2	pr = 0.3	1031,0	1035,0	10581.00	10639.50	125,0	129,5	123,0	133,0	126,0	128,5
	pr = 0.5	1035,0	1036,0	10475.00	10741.50	123,0	134,0	109,0	125,0	122,0	122,0
	pr = 0.7	1045,0	1051,0	10348.00	10730.50	111,0	116,0	109,0	120,0	127,0	130,0
pop = 100 (ger = 2500) pm = 100 pr = 0.7 tsize = 2	pm = 0.0	1037,0	1043,5	10137.00	10347.50	201,0	234.50	189,0	197,5	313,0	320,5
	pm = 0.001	1032,0	1033,0	10690.00	10867.50	122,0	123,0	118,0	145,5	127,0	127,0
	pm = 0.01	1033,0	1041,0	10865.00	10948.50	121,0	121,5	114,0	117,0	136,0	140,5
pr = 0.7 pm = 0.05 tsize = 2	pm = 0.05	1026,0	1029,0	10497.00	10766.0	129,0	135.00	107,0	108,5	185,0	188,5
	pop = 10 (ger = 7.5K)	1025.00	1029.50	10110.0	10210.0	120,0	121,5	108,0	117,0	117,0	120,0
	pop = 50 (ger = 5K)	1036,0	1038.50	10209.0	10417.5	125,0	130,0	107,0	109,0	139,0	145,0
pm = 0.05 tsize = 2	pop = 100 (ger = 2.5K)	1029,0	1034.0	10265.0	10501.5	123,0	132.5	114,0	114,0	172,5	165,0
Parâmetros Fixos	Parâmetros a variar	Algoritmo base		com penalização		com reparação		com mutação por troca + reparação		com recombinação uniforme + mutação binária + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (gen = 25) pr = 0.7 pm = 0.001	tsize = 3	1028,0	1030,0	10167.0	10195.0	133,0	135,0	151,0	164,0	119,0	119,5
	tsize = 10	1028,0	1030,0	10889.0	11105.5	107,0	107,0	242,0	242,0	121,0	123,0
	tsize = 50	1038,0	1039.50	10834.0	10934.5	107,0	107,0	117,0	134,0	116,0	120,0

**TABELA 9 - Algoritmo Evolutivo com Variações de Operadores (Recombinação, Mutação, Penalização, Reparação) e Parâmetros (Probabilidade de Recombinação, Probabilidade de Mutação, Tamanho da População) no file5.txt**

Ficheiro file5.txt											
Parâmetros Fixos	Parâmetros a variar	Algoritmo base (Recombinação de 1 ponto de corte + Mutação + Penalização cega)		Recombinação de 1 ponto de corte + Mutação + Penalização não cega		Recombinação de 1 ponto de corte + Mutação + Reparação		Recombinação de 1 ponto de corte + Mutação por troca + Reparação		com recombinação uniforme + mutação + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (ger = 2500) pm = 0.01 tsize = 2	pr = 0.3	1042,0	1045,0	2754,00	2815,50	4,0	4,0	4,0	4,0	4,0	4,0
	pr = 0.5	1041,0	1043,0	2507,00	2985,50	4,0	4,0	4,0	4,0	4,0	4,0
	pr = 0.7	1059,0	1060,5	2630,00	2749,00	4,0	4,0	4,0	4,0	4,0	4,0
pop = 100 (ger = 2500) pop = 100 pr = 0.7 tsize = 2	pm = 0.0	1040,0	1041,0	2911,00	3196,0	4,0	4,0	4,0	4,0	4,0	4,0
	pm = 0.001	1025,0	1035,5	3886,00	4101,0	4,0	4,0	4,0	4,0	4,0	4,0
	pm = 0.01	1032,0	1037,0	2599,00	3213,0	4,0	4,0	4,0	4,0	4,0	4,0
pr = 0.7 pm = 0.05 tsize = 2	pm = 0.05	1043,0	1049,0	2717,00	3315,5	4,0	4,0	4,0	4,0	4,0	4,0
	pop = 10 (ger = 7.5K)	1027,00	1029,00	2146,0	2319,5	4,0	4,0	4,0	4,0	4,0	4,0
	pop = 50 (ger = 5K)	1028,0	1031,00	3012,00	3242,5	4,0	4,0	4,0	4,0	4,0	4,0
pop = 100 (ger = 2.5K)	pop = 100 (ger = 2.5K)	1034,00	1034,5	2196,0	2992,5	4,0	4,0	4,0	4,0	4,0	4,0
Parâmetros Fixos	Parâmetros a variar	Algoritmo base		com penalização		com reparação		com mutação por troca + reparação		com recombinação uniforme + mutação binária + reparação	
		Best	MBF	Best	MBF	Best	MBF	Best	MBF	Best	MBF
pop = 100 (ger = 25) pr = 0.7 pm = 0.001	tsize = 3	1042,0	1044,0	2331,0	2443,0	4,0	4,0	4,0	4,0	4,0	4,0
	tsize = 10	1043,0	1043,0	3724,00	3818,5	4,0	4,0	4,0	4,0	4,0	4,0
	tsize = 50	1034,0	1044,0	2775,0	3200,0	4,0	4,0	4,0	4,0	4,0	4,0

## Conclusão Geral sobre o Algoritmo Evolutivo

O algoritmo evolutivo mostrou ser uma abordagem superior ao algoritmo de Trepa-Colinas em várias situações, principalmente pela sua capacidade de explorar um espaço de soluções maior e de “escapar” de ótimos locais, algo que o Trepa-Colinas tem dificuldades. Enquanto o Trepa-Colinas realiza uma busca local, ajustando-se a soluções vizinhas sem considerar alternativas distantes, o algoritmo evolutivo utiliza uma população de soluções e mecanismos de cruzamento e mutação, permitindo uma exploração maior e a possibilidade de encontrar soluções globais de maior qualidade.

No entanto, houve uma configuração do algoritmo evolutivo que se destacou, tanto de forma negativa quanto positiva. Quando os parâmetros de taxa de mutação estavam muito altos, o algoritmo teve um desempenho pior, já que as mutações excessivas geraram soluções aleatórias, sem um bom aproveitamento do conhecimento acumulado pelas gerações anteriores. Por outro lado, com uma taxa de mutação moderada e uma boa configuração da população inicial, o algoritmo teve um desempenho muito satisfatório, encontrando soluções mais robustas e melhores do que as obtidas pelo Trepa-Colinas, especialmente em problemas mais complexos.

Portanto, o algoritmo evolutivo apresenta vantagens claras em termos de exploração do espaço de soluções, mas exige um ajuste cuidadoso de parâmetros para garantir um bom desempenho

**TABELA 10 - Algoritmo Híbrido com Variações de probabilidade de gerar vizinho no file1.txt**

Parâmetros Fixos	Parâmetros a variar	Algoritmo base híbrido i)		Algoritmo base híbrido ii)	
		Best	MBF	Best	MBF
pop = 100 (gen = 2.5k)	PROBGERAVIZ = 1	2,0	2,0	2,0	2,0
pr = 0.7	PROBGERAVIZ = 0.8	2,0	2,0	2,0	2,0
pm = 0.001					
tsize = 2					

**TABELA 11 - Algoritmo Híbrido com Variações de probabilidade de gerar vizinho no file2.txt**

Parâmetros Fixos	Parâmetros a variar	Algoritmo base híbrido i)		Algoritmo base híbrido ii)	
		Best	MBF	Best	MBF
pop = 100 (gen = 2.5k)	PROBGERAVIZ = 1	20,0	20,0	20,0	20,0
pr = 0.7	PROBGERAVIZ = 0.8	20,0	20,0	20,0	20,0
pm = 0.001					
tsize = 2					

**TABELA 12 - Algoritmo Híbrido com Variações de probabilidade de gerar vizinho no file3.txt**

Parâmetros Fixos	Parâmetros a variar	Algoritmo base híbrido i)		Algoritmo base híbrido ii)	
		Best	MBF	Best	MBF
pop = 100 (gen = 2.5 pr = 0.7 pm = 0.001 tsize = 2	PROBGERAVIZ = 1	102,0	102,0	102,0	103,0
	PROBGERAVIZ = 0.8	102,0	102,0	102,0	102,0

**TABELA 13 - Algoritmo Híbrido com Variações de probabilidade de gerar vizinho no file4.txt**

Parâmetros Fixos	Parâmetros a variar	Algoritmo base híbrido i)		Algoritmo base híbrido ii)	
		Best	MBF	Best	MBF
pop = 100 (gen = 2.5 pr = 0.7 pm = 0.001 tsize = 2	PROBGERAVIZ = 1	129,0	134,5	120,0	120,5
	PROBGERAVIZ = 0.8	109,0	116,0	112,0	114,0

**TABELA 14 - Algoritmo Híbrido com Variações de probabilidade de gerar vizinho no file5.txt**

Parâmetros Fixos	Parâmetros a variar	Algoritmo base híbrido i)		Algoritmo base híbrido ii)	
		Best	MBF	Best	MBF
pop = 100 (gen = 2.5 pr = 0.7 pm = 0.001 tsize = 2	PROBGERAVIZ = 1	4,0	4,0	4,0	4,0
	PROBGERAVIZ = 0.8	4,0	4,0	4,0	4,0

## Conclusão Geral sobre o Algoritmo Híbrido

O algoritmo híbrido apresentou um desempenho superior em relação ao algoritmo evolutivo e ao trepa colinas. Ele foi capaz de alcançar melhores resultados em termos de solução ótima, devido à combinação das técnicas de busca que permitem uma exploração mais eficaz do espaço de soluções. Em comparação, o algoritmo evolutivo teve um desempenho mais limitado, enquanto o trepa colinas, por ser mais simples, teve dificuldades em escapar de soluções locais. O híbrido, ao integrar a

geração de vizinhos, mostrou-se mais robusto e eficaz, destacando-se como a melhor abordagem entre as três.

## **5. CONCLUSÃO**

Neste trabalho, implementaram-se e testaram-se diferentes abordagens para resolver o problema da combinação de moedas. O uso de algoritmos evolutivos, trepa-colinas e combinações híbridas permitiu explorar e refinar soluções de maneira eficaz. Os resultados mostram que a abordagem híbrida, particularmente a que aplica o trepa-colinas ao melhor indivíduo de cada geração, oferece um bom equilíbrio entre exploração global e refinamento local. Trabalhos futuros podem explorar melhorias no operador de reparo e na definição das vizinhanças.



**Instituto Superior  
de Engenharia**

Politécnico de Coimbra