Instituto Superior
de Engenharia
Politécnico de Coimbra

Departamento de Engenharia Informática e de Sistemas

Trabalho Prático nº 2
Problema de Otimização
Coin Change Problem

Trabalho de Projeto para a Unidade Curricular de Introdução à Inteligência Artificial

Dario José Ferreira Santos **LEI 2021110772**

José Manuel Bastos Correia **LEI 2021127160**

Coimbra, dezembro 2024

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

TABELAS

1 INTRODUÇÃO

Neste relatório vamos abordar a resolução do *Coin Change Problem*, um clássico problema de otimização.

O objetivo é determinar a combinação de moedas que permita alcançar um valor V com o menor número possível de moedas. Para tal, foram implementados e analisados três métodos de otimização: pesquisa local, algoritmo evolutivo e duas abordagens híbridas que combinam os dois métodos anteriores. A análise foca-se em estratégias para lidar com soluções inválidas, bem como na exploração de diferentes parâmetros e operadores de cada algoritmo.

As instâncias testadas apresentam diferentes níveis de complexidade, o que ajuda a avaliar a eficiência das soluções propostas. O trabalho visa comparar resultados, tentando destacar as melhores abordagens para cada método.

2 CONFIGURAÇÃO

Para cumprir com a devida análise de resultados são inicialmente configurados os parâmetros das funções.

```
Solution local_search(CoinChangeProblem *problem, int max_iterations);
Solution evolutionary_algorithm(CoinChangeProblem *problem, int
population_size, int generations);
Solution hybrid_algorithm(CoinChangeProblem *problem, int max_iterations,
int population_size, int generations);
```

- Max_iterations;
- Population_size;
- Generations;

Após execução é feita a avaliação com critério d *output* em cada execução, para cada variação o processo é executado 10 vezes, ao concluir o processo são impressos os resultados individualmente, juntamente com o cálculo da média e o melhor resultado encontrado (total de moedas).

Local Search Solutions:

Run: 1 // Executado 10 vezes;

Coin Value 0.05: 0

Coin Value 0.20: 0

Coin Value 0.50: 2

Total Coins: 2

Total Value: 1.00

// Após 10 execuções;

Local Search Best Total Coins: 2

Local Search Average Total Coins: 5.30

O mesmo processo é executado para o evolutivo e híbrido

2.1 Modelos

Configurações :

- Runs: 10;
- Ficheiros: “file1.txt”, “file2.txt”, “file3.txt”, “file4.txt”, “file5.txt”;
- Iterações: 100, 1000, 5000, 10000;
- População: 50, 100, 250, 500;
- Gerações: 100;
- Mutações: Incremental, Troca;
- Crossover: *Single Point*, *Uniform*;

2.2 Modelos Base

2.2.1 Pesquisa Local

Pesquisa Local		100it	1000	5000	10000
File1.txt	**Best**	2	2	3	2
	AVG	4,4	5,6	7,4	5,9
File2.txt	**Best**	37	35	38	39
	AVG	50,8	50,2	46,9	50
File3.txt	**Best**	164	183	177	179
	AVG	188,2	190,6	187,6	194,5
File4.txt	**Best**	256	363	356	356
	AVG	269,3	384,5	395,9	386
File5.txt	**Best**	4	6	4	4
	AVG	11	9,9	6,9	5,6

Table 1 Pesquisa Local Iterações

A Pesquisa Local aparenta ser eficiente apenas em problemas simples, apresentando limitações na apresentação de soluções com maior qualidade em problemas complexos.

2.2.2 Evolutivo

Evolutivo		100it	1000	5000	10000
File1.txt	**Best**	2	2	2	2
	AVG	3,5	4,4	4,1	3,8
File2.txt	**Best**	42	43	43	45
	AVG	47	47,3	48	48,2
File3.txt	**Best**	103	194	97	101
	AVG	105,2	106,6	107,3	108,2
File4.txt	**Best**	140	147	140	144
	AVG	157,3	161	162,3	162,6
File5.txt	**Best**	4	4	4	4
	AVG	6,5	6,9	6,3	6,3

Table 2 Evolutivo - Iterações

O algoritmo Evolutivo parece ser o mais robusto em termos de estabilidade e qualidade geral das soluções.

2.2.3 Híbrido

Híbrido		100it	1000	5000	10000
File1.txt	**Best**	2	2	2	2
	AVG	6,2	5,3	4,1	3,8
File2.txt	**Best**	41	35	38	39
	AVG	52,7	51,2	50,4	49,6
File3.txt	**Best**	178	173	166	183
	AVG	185,8	185,1	186,9	188,6
File4.txt	**Best**	347	373	362	363
	AVG	382,8	401,8	387,7	394,3
File5.txt	**Best**	5	4	4	4
	AVG	4,9	7,3	6,9	6,7

Table 3 Híbrido Base

O Método Híbrido é competitivo em instâncias simples, mas não apresenta melhorias significativas em problemas mais complexos.

2.3 Configurações

2.3.1 População

Para otimizar o algorítmo híbrido fomos descobrir a melhor quantidade de população para aplicar na melhoria dos próximos métodos no algorítmo Evolutivo.

Assim sendo, foi executado o algoritmo evolutivo com:

- Iterações: 10000;
- Gerações 100;

Table 4 Evolutivo -População

Evolutivo		50 pop	100	250	500
File1.txt	**Best**	2	2	2	2
	AVG	4,1	3,2	2,9	2,9
File2.txt	**Best**	46	46	45	44
	AVG	47,7	47,6	47,4	46,6
File3.txt	**Best**	106	106	113	121
	AVG	108,6	116,1	121,6	125,8
File4.txt	**Best**	141	163	176	174
	AVG	158	174,9	189,7	190,1
File5.txt	**Best**	4	4	4	4
	AVG	6,3	5,7	5	4,9

Com a tabela acima podemos concluir que o algorítmo evolutivo com baixa população tem bastante eficácia quando trata de problemas complexos.

Já quando a popução é mais elevada os problemas menos complexos tendem a melhorar ligeiramente.

2.3.2 Mutação e Crossover

Para melhorar o algorítmo híbrido ainda foram estudadas duas variações de mutação e crossover.

Para este estudo estabilizamos os parametros de:

- Iterações: 10000;
- Gerações:100;
- População: 500;

Esta análise serve para perceber o impacto de cada variação de mutação e crossover.

A mutação foi isolada para apenas fazer mutação incremental ou por troca e o crossover para apenas fazer ponto único ou uniforme.

Inicialmente estavam os dois ambos com um probabilidade de 50% de executar, assim pode-mos focar probabilidade de execução para obter melhores resultados.

Mutação			Crossover	
Incremental	Swap		Single Point	Uniform
2	5		2	2
4,1	5		2,3	2,9
33	40		45	47
34,6	40		46,5	47,1
132	155		187	122
138,1	160		194,1	127,1
158	330		167	177
169	388,2		189	198,9
5	4		4	4
8,6	4		5,7	4,6

Table 5 Mutação e Crossover

Para otimizar podemos colocar uma chance de Mutaçao Incremental de 80% e 20% de Troca.

Para otimizar podemos colocar uma chance de Crossover Ponto Único de 60% e 40% de Uniforme.

2.4 Otimização

Realizá-mos então o estudo do híbrido para descobrir como podemos filtrar os melhores resultados através da configuração das iterações, mantendo o população no máximo para cada ficheiro avaliado, assim como os dois tipos de mutação e os dois tipos de crossover com uma probabilidade de 50% cada.

Híbrido		100it	1000	5000	10000
File1.txt	**Best**	2	2	2	2
	AVG	6,2	5,3	4,1	3,8
File2.txt	**Best**	41	35	38	39
	AVG	52,7	51,2	50,4	49,6
File3.txt	**Best**	178	173	166	183
	AVG	185,8	185,1	186,9	188,6
File4.txt	**Best**	347	373	362	363
	AVG	382,8	401,8	387,7	394,3
File5.txt	**Best**	5	4	4	4
	AVG	4,9	7,3	6,9	6,7

Table 6 Híbrido - Iterações

A tabela apresentada confirma que o algoritmo híbrido melhora consistentemente na obtenção da melhor solução à medida que mais iterações são executadas.

Para aprimorar ainda mais a eficiência do método híbrido, ajustamos as probabilidades associadas a cada tipo de mutação e crossover, utilizando os valores mais eficazes observados em testes preliminares.

O objetivo desta abordagem foi demonstrar que o algoritmo híbrido pode ser otimizado para diferentes conjuntos de dados ao diversificar as probabilidades de execução de cada operador de mutação e crossover. Para este experimento, foram aplicadas as seguintes probabilidades:Foram aplicadas então as seguintes probabilidades de execução:

- Mutação: Incremental (80%) e Troca (20%);
- Crossover: Ponto único (60%), Uniforme (40%);

3 CONCLUSÃO

Híbrido Otimizado		Híbrido Mut/Cross 50/50	Híbrido 1 Mut/Cross 80/20; 60/40
File1.txt	**Best**	2	2
	AVG	4,1	3,8
File2.txt	**Best**	44	48
	AVG	50	54
File3.txt	**Best**	188	169
	AVG	192,6	186,7
File4.txt	**Best**	370	362
	AVG	394,3	388,1
File5.txt	**Best**	4	4
	AVG	7,3	6,6

Table 7 Híbrido Otimizado

De forma geral, o Híbrido (Mut/Cross 80/20; 60/40) revelou-se mais eficiente e consistente do que o Híbrido (Mut/Cross 50/50), apresentando melhores valores médios e soluções mais robustas em grande parte dos casos.

Embora em alguns ficheiros os melhores valores tenham sido semelhantes ou ligeiramente superiores no Híbrido (50/50), a diversificação das probabilidades de mutação e crossover no Híbrido (80/20; 60/40) permitiu alcançar um desempenho global mais estável e otimizado.

Instituto Superior
de Engenharia
Politécnico de Coimbra