

Heurísticas para um problema NP-Difícil

**Gabriel P. Soares¹, Lucas P. O. Alves¹, Leticia F. Oliveira¹,
Gustavo F. Pascoaleto¹, Jorran L. A. Santos¹ Pedro H. de Almeida¹**

¹ Instituto de Ciências Exatas – Ciência da Computação
Universidade Federal de Alfenas (UNIFAL)
Junho de 2024

1. Introdução

Este relatório apresenta o desenvolvimento e a comparação de duas heurísticas para resolver o problema de Soma de Subconjuntos, que é um problema NP-Difícil. O objetivo é analisar a eficácia dessas abordagens na busca de soluções eficientes para problemas complexos.

O problema de Soma de Subconjuntos consiste em determinar se existe um subconjunto de um conjunto dado de números inteiros cuja soma é igual a um valor alvo específico. Este problema é conhecido por sua complexidade computacional e por ser um exemplo clássico de problema NP-Difícil, desafiando as capacidades de algoritmos tradicionais.

Para abordar esse problema, implementamos duas heurísticas diferentes: um algoritmo de busca local e um algoritmo evolutivo. O algoritmo de busca local explora soluções vizinhas para encontrar a melhor solução possível em um determinado espaço de busca. O algoritmo evolutivo, por sua vez, utiliza princípios de seleção natural, cruzamento e mutação para evoluir uma população de soluções ao longo de várias gerações.

A comparação entre as heurísticas será feita com base em dois critérios principais: tempo de execução e qualidade das soluções encontradas. Para garantir uma avaliação abrangente, utilizamos dez instâncias de teste diferentes, obtidas de repositórios online ou desenvolvidas pelos membros do grupo.

Este relatório está organizado da seguinte maneira: primeiramente, descrevemos o problema de Soma de Subconjuntos em detalhes. Em seguida, explicamos as heurísticas implementadas e a metodologia de comparação utilizada. Por fim, apresentamos e discutimos os resultados obtidos, destacando a eficácia de cada abordagem e as conclusões derivadas dos experimentos.

2. Algoritmos

Algoritmo Evolutivo

O algoritmo evolutivo é uma heurística inspirada nos princípios da evolução natural, onde uma população de soluções evolui ao longo de várias gerações para encontrar uma solução otimizada para um problema. Neste caso, o objetivo é encontrar um subconjunto de números cuja soma seja o mais próximo possível de um valor alvo. O algoritmo utiliza operações de seleção, cruzamento e mutação para melhorar a qualidade das soluções ao longo do tempo.

Inicialização:

Gerar uma população inicial de soluções aleatórias.

Avaliação:

Calcular a aptidão (fitness) de cada indivíduo na população.

Seleção:

Selecionar pares de indivíduos (pais) com base em suas aptidões, utilizando uma roleta viciada.

Cruzamento:

Gerar novos indivíduos (filhos) combinando partes dos pais selecionados.

Mutação:

Aplicar mutações aleatórias aos filhos para introduzir variação.

Substituição:

Formar uma nova população a partir dos filhos gerados.

Repetição:

Repetir os passos de avaliação, seleção, cruzamento e mutação por um número definido de gerações.

Terminação:

Retornar o melhor indivíduo encontrado ao final das gerações.

Algoritmo de Busca Local

O algoritmo de busca local é uma heurística que busca encontrar uma solução melhor a partir de uma solução inicial, explorando as soluções vizinhas. O objetivo é encontrar um subconjunto de números cuja soma seja o mais próximo possível de um valor alvo. O algoritmo realiza pequenas mudanças na solução atual para explorar novas soluções e melhorar a qualidade da solução encontrada.

Inicialização:

Definir uma solução inicial aleatória.

Avaliação:

Calcular a aptidão (fitness) da solução inicial.

Busca Local:

Repetir por um número definido de iterações:

- Gerar uma nova solução vizinha invertendo um bit aleatório da solução atual. - Calcular a aptidão da nova solução. - Se a nova solução for melhor, atualizá-la como a melhor solução encontrada.

Terminação:

Retornar a melhor solução encontrada após o número definido de iterações.

Descrição das Instâncias

Para avaliar o desempenho das heurísticas implementadas, foram propostas diferentes instâncias de teste. Estas instâncias foram escolhidas para representar uma variedade de cenários, com conjuntos de números de diferentes tamanhos e distribuições, permitindo uma avaliação abrangente das heurísticas. Abaixo, segue a descrição detalhada de cada uma das instâncias:

Conjunto 1 original:

1, 2, 3, 5, 7, 10, 12, 14, 15, 18, 20, 22, 25, 27, 30, 35, 40, 45, 50, 55

Conjunto 2: Pequeno e simples cpp 1, 3, 5, 7, 9, 11, 13, 15, 17, 19

- Tamanho: 10 elementos. - Características: Valores ímpares em sequência. - Propósito: Testar a eficiência do algoritmo em um conjunto pequeno e relativamente uniforme.

Conjunto 3: Pequeno com variação de valores cpp 2, 5, 8, 14, 21, 34, 55, 89, 144, 233

- Tamanho: 10 elementos. - Características: Sequência que cresce rapidamente (valores não são uniformemente espaçados). - Propósito: Avaliar como o algoritmo lida com valores que têm uma variação significativa.

Conjunto 4: Médio e uniforme cpp 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150

- Tamanho: 15 elementos. - Características: Valores múltiplos de 10. - Propósito: Testar o desempenho do algoritmo em um conjunto de tamanho médio com valores uniformemente espaçados.

Conjunto 5: Médio com variação cpp 3, 6, 9, 27, 54, 108, 216, 432, 864, 1728, 3456, 6912, 13824, 27648, 55296

- Tamanho: 15 elementos. - Características: Crescimento exponencial. - Propósito: Verificar o comportamento do algoritmo quando os valores crescem exponencialmente.

Conjunto 6: Grande e aleatório cpp 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

- Tamanho: 25 elementos. - Características: Números primos. - Propósito: Testar a eficiência do algoritmo em um conjunto maior com valores não uniformes e imprevisíveis.

Conjunto 7: Grande com números grandes cpp 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900, 2000

- Tamanho: 20 elementos. - Características: Valores grandes e uniformemente espaçados. - Propósito: Avaliar como o algoritmo se comporta com valores grandes e uniformemente espaçados.

Conjunto 8: Sequência de Fibonacci cpp 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765

- Tamanho: 20 elementos. - Características: Sequência de Fibonacci. - Propósito: Testar a eficiência do algoritmo com uma sequência que cresce rapidamente, mas de forma não exponencial.

Conjunto 9: Potências de 2 cpp 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288

- Tamanho: 20 elementos. - Características: Potências de 2. - Propósito: Avaliar o desempenho do algoritmo em uma sequência exponencial.

Conjunto 10: Conjunto misto com números pequenos e grandes cpp 3, 6, 9, 15, 21, 33, 48, 75, 123, 195, 312, 501, 813, 1314, 2127, 3441, 5568, 9009, 14577, 23688

- Tamanho: 20 elementos. - Características: Combinação de valores pequenos e grandes. - Propósito: Testar a robustez do algoritmo em um conjunto misto com uma grande variação de valores.

Esses conjuntos foram escolhidos para cobrir uma ampla gama de situações que o algoritmo pode encontrar. Eles variam em tamanho, tipo de valores (uniformes, aleatórios, exponenciais) e complexidade, proporcionando um bom teste da eficiência e eficácia da heurística implementada.

3. Resultados

Valor da soma: 100

Algoritmo Busca Local:

Conjunto 1:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 2

Melhor subconjunto encontrado: 7 10 15 25 45

Tempo de execução: 2420 microsegundos

O valor da soma de subconjuntos escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 2:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 1 3 5 7 9 11 13 15 17 19

Tempo de execução: 1772 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 3:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 44

Melhor subconjunto encontrado: 144

Tempo de execução: 2329 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 4:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 10 20 70

Tempo de execução: 1545 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 5:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 8

Melhor subconjunto encontrado: 108

Tempo de execução: 3060 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 6:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 2 3 5 89

Tempo de execução: 2518 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 7:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 100

Tempo de execução: 1255 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 8:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 1 2 8 89

Tempo de execução: 1095 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 9:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 28

Melhor subconjunto encontrado: 128

Tempo de execução: 1916 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Conjunto 10:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 95

Melhor subconjunto encontrado: 195

Tempo de execução: 2096 microsegundos

O valor da soma de subconjunto escolhido foi: 100

O número de interações escolhido foi: 2000

Valor da soma: 54**Conjunto 1:**

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 10 45

Tempo de execução: 1001 microsegundos

O valor da soma de subconjuntos escolhido foi: 54

O número de interações escolhido foi: 1500

Conjunto 2:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 9 11 15 19

Tempo de execução: 0 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 1500

Conjunto 3:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 35

Melhor subconjunto encontrado: 89

Tempo de execução: 0 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 1500

Conjunto 4:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 26

Melhor subconjunto encontrado: 80

Tempo de execução: 1000 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 1500

Conjunto 5:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 54

Tempo de execução: 999 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 1500

Conjunto 6:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 53

Tempo de execução: 999 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 1500

Conjunto 7:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 46

Melhor subconjunto encontrado: 100

Tempo de execução: 1002 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 1500

Conjunto 8:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 55

Tempo de execução: 999 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 1500

Conjunto 9:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 1 4 16 32

Tempo de execução: 2001 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 2000

Conjunto 10:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 21 33

Tempo de execução: 0 microsegundos

O valor da soma de subconjunto escolhido foi: 54

O número de interações escolhido foi: 1500

Valor da soma: 100

Algoritmo Busca Evolutiva:

Conjunto 1:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 1 5 12 22 25 35

Tempo de execução: 807 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 2:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 1 3 5 7 9 11 13 15 17 19

Tempo de execução: 643 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 3:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 2 8 14 21 55

Tempo de execução: 334 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 4:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 10 20 30 40

Tempo de execução: 712 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 5:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 3 6 9 27 54

Tempo de execução: 302901 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 6:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 5 11 23 61

Tempo de execução: 1898 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 7:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 100

Tempo de execução: 3416 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 8:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 2

Melhor subconjunto encontrado: 13 34 55

Tempo de execução: 319420 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 9:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 5

Melhor subconjunto encontrado: 1 2 4 8 16 64

Tempo de execução: 316566 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Conjunto 10:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 6 9 15 21 48

Tempo de execução: 313360 microsegundos

O valor da soma de subconjunto escolhido foi: 100

Valor da soma: 54**Conjunto 1:**

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 2 5 10 15 22

Tempo de execução: 1001 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 2:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 1 5 9 11 13 15

Tempo de execução: 1001 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 3:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 55

Tempo de execução: 329074 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 4:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 4

Melhor subconjunto encontrado: 20 30

Tempo de execução: 346079 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 5:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 54

Tempo de execução: 4001 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 6:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 5 7 11 31

Tempo de execução: 4001 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 7:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 46

Melhor subconjunto encontrado: 100

Tempo de execução: 365083 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 8:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 1

Melhor subconjunto encontrado: 55

Tempo de execução: 370085 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 9:

Não foi possível encontrar um subconjunto que soma exatamente ao valor alvo.

A solução mais próxima encontrada tem uma diferença de: 10

Melhor subconjunto encontrado: 64

Tempo de execução: 369083 microsegundos

O valor da soma de subconjunto escolhido foi: 54

Conjunto 10:

Encontrou um subconjunto que soma exatamente ao valor alvo.

Melhor subconjunto encontrado: 6 48

Tempo de execução: 4001 microsegundos

O valor da soma de subconjunto escolhido foi: 54

É perceptível, pelos resultados, que o algoritmo de Busca Local obteve um melhor desempenho.

Algoritmo Busca Local	
Conjunto	Tempo (μs)
Conjunto 1	2420
Conjunto 2	1772
Conjunto 3	2329
Conjunto 4	1545
Conjunto 5	3060
Conjunto 6	2518
Conjunto 7	1255
Conjunto 8	1095
Conjunto 9	1916
Conjunto 10	2096

Figura 1. Tabela Busca Local

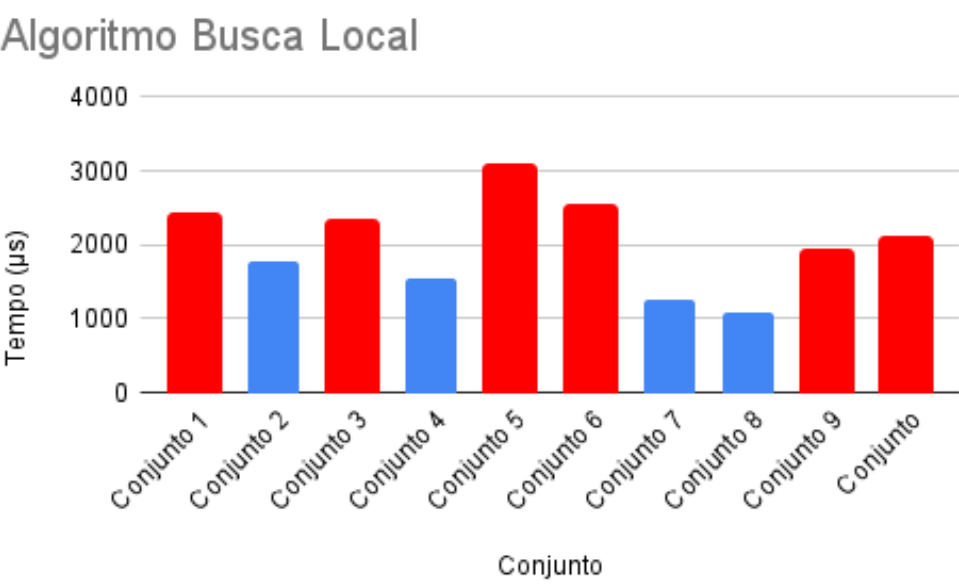


Figura 2. Gráfico de Busca Local

Algoritmo Busca Evolutiva	
Conjunto	Tempo (μs)
Conjunto 1	807
Conjunto 2	643
Conjunto 3	334
Conjunto 4	712
Conjunto 5	302901
Conjunto 6	1898
Conjunto 7	3416
Conjunto 8	319420
Conjunto 9	316566
Conjunto 10	313360

Figura 3. Enter Caption

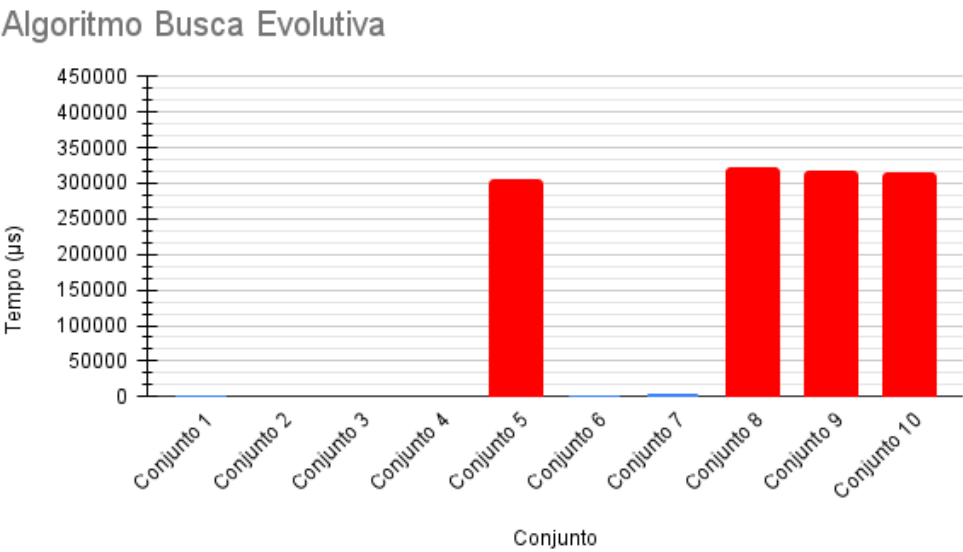


Figura 4. Enter Caption