Prova da NP-completude do Problema da Partição de Números (TWNPP)

Guilherme Oliveira Araujo, João Pedro Carvalho Ferreira, Luis Renato Goulart, Enzo Maranezi,Mileno Oliveira Matos,Isabella Cristina da Silveira

[1]Departamento de Ciências da Computação – Universidade Federal de Alfenas
Avenida Jovino Fernandes de Sales 2600 – CEP 37133840 - Alfenas – MG - Brasil

guilherme.araujo1@sou.unifal-mg.edu.br,joaopedro.carvalho@sou.unifal-mg.edu.br,luis.goulat@sou.unifal-mg.edu.br,enzo.maranezi@sou.unifal-mg.edu.br,mileno.matos@sou.unifal-mg.edu.br,isabella.cristina@sou.unifal-mg.edu.br

Resumo.O Problema da Partição de Números (TWNPP) é um problema clássico da complexidade computacional, classificado como NP-completo. Ele consiste em dividir um conjunto de números inteiros em dois subconjuntos com somas iguais. Para provar sua NP-completude, foi utilizada a redução polinomial a partir do Problema da Soma de Subconjuntos, também NP-completo. Os resultados confirmam que a verificação de uma solução pode ser feita em tempo polinomial, mas encontrar a solução exata é exponencialmente difícil. Testes comparativos entre abordagens exatas e heurísticas demonstram que métodos aproximados são mais viáveis para grandes instâncias.

1. Introdução

O problema da partição de números (Two-Way Number Partitioning Problem - TWNPP) é um dos problemas clássicos da teoria da complexidade computacional e tem aplicações diretas em otimização, balanceamento de carga e alocação de recursos. Ele pertence à classe NP-completo e pode ser descrito da seguinte forma: dado um conjunto de números inteiros positivos, existe uma maneira de dividi-lo em dois subconjuntos cuja soma dos elementos seja igual? Esse problema foi formalmente identificado como NP-completo em 1972 por Richard Karp, sendo um dos 21 problemas fundamentais dessa classe. A importância desta classificação reside no fato de que, se houvesse uma solução eficiente para este problema, então uma grande quantidade de outros problemas NP-completos também poderiam ser resolvidos eficientemente. A metodologia utilizada neste estudo envolve a definição formal do Problema da Partição de Números e do Problema do Subconjunto da soma, seguidas das respectivas provas de pertencimento à classe NP e da prova de NP-completude por meio da redução polinomial. Primeiramente, demonstramos que o Problema da Participação de Números pertence a NP, mostrando que uma solução candidata pode ser verificada em tempo polinomial. Em seguida, provamos a NP-completude do problema, transformando uma instância do Problema do Subconjunto da soma em uma instância equivalente do Problema da Partição de Números em tempo polinomial. Os principais resultados alcançados incluem a confirmação da classificação NP-completa do Problema da Partição, demonstrando que a verificação de uma solução pode ser feita de forma eficiente, mas que encontrar a solução em si permanece um desafio computacional significativo.

2.Metodologia

2.1.Problema da Partição de Números(Problema a ser provado)

O **Problema da Partição de Números (Two-Way Number Partitioning Problem – TWNPP)** é um problema de otimização combinatória e teoria da complexidade computacional. Ele consiste em determinar se um conjunto de números inteiros pode ser dividido em dois subconjuntos cuja soma dos elementos seja igual. A definição formal diz que, dado um conjunto de n números inteiros positivos S={S1,S2, ..,Sn}, o problema da partição pergunta se existe um subconjunto S1 ⊆ S tal que a soma de seus elementos seja exatamente a metade da soma total dos elementos do conjunto original, ou seja, queremos encontrar dois subconjuntos S1 e S2 disjuntos que satisfaçam:

$$\sum_{s \in S_1} s = \sum_{s \in S_2} s$$

Para melhor entendimento vamos exemplificar a ideia.

A sequência V = {87, 6, 5, 45, 34, 2, 24, 12, 7, 6, 54, 34} pode ser particionada da seguinte forma:

$$\underbrace{\{87, 34, 24, 6, 5\}}_{\sum_{i \in A_1} v_i = 156}, \underbrace{\{54, 45, 34, 12, 7, 6, 2\}}_{\sum_{i \in A_2} v_i = 160}$$

O valor da função dessa partição é 160−156 = 4, mas a partição ótima seria:

$$\underbrace{\{87, 34, 24, 6, 5, 2\}}_{\sum_{i \in A_1} v_i = 158}, \underbrace{\{54, 45, 34, 12, 7, 6\}}_{\sum_{i \in A_2} v_i = 158}$$

com valor da função 158 – 158 = 0.

O enunciado matemático do Problema da Partição de Números consiste em encontrar um conjunto A ≠ ∅, tal que A ⊂ In, que minimize a função:

$$f(A) = \left| \sum_{i \in A} v_i - \sum_{i \in I_n \setminus A} v_i \right|$$

2.2 Problema da Soma de Subconjuntos (Problema para a Prova de NP-Completude)

O **Problema da Soma de Subconjuntos (Subset Sum Problem)** é um problema da teoria da complexidade computacional e da otimização combinatória. Ele pertence à classe dos problemas de decisão e consiste em determinar se existe um subconjunto de um conjunto dado de números inteiros cuja soma seja igual a um valor alvo específico.

A definição formal diz que, dado um conjunto de números inteiros **S** e um número inteiro alvo **T**, o problema pergunta se existe um subconjunto **S'** $\subseteq$ **S** tal que:

$$\sum_{s \in S'} s = T$$

Se tal subconjunto existir, a resposta é **"Sim"**, caso contrário, a resposta é **"Não"**.

Dado um vetor de números inteiros **p[1..n]** e um subconjunto **X** de $\{1, 2, \dots, n\}$, denotaremos por **p(X)** a soma $\sum_{i \in x} p[i]$. Essa notação é útil para formalizar o seguinte problema.

Os elementos de **p[1..n]** são chamados pesos e o parâmetro **"T"** é chamado alvo. O problema consiste em decidir se algum subconjunto de **[1..n]** tem soma dos pesos igual ao alvo.

Toda instância do problema tem solução. Uma solução **"Sim"** pode ser representada por **1** e uma solução **"não"** por **0**. Uma instância com **n = 0** tem solução sim se e somente se o alvo é **T=0**.

Embora o problema peça apenas sim ou não como resposta, qualquer algoritmo para o problema pode ser modificado para produzir, junto com um sim, um conjunto **X** tal que **p(X) = T**.

Exemplo : A instância do problema que tem pesos p = (30, 40, 10, 15, 10, 60, 54) e alvo T= 55 tem solução sim. O conjunto $\{2, 4\}$ tem soma de pesos igual ao alvo. O conjunto $\{1, 4, 5\}$ também tem essa propriedade.

	1	2	3	4	5	6	7
p	30	40	10	15	10	60	54
		☑		☑			
	☑			☑	☑		

O **Problema da Soma de Subconjuntos** é sabido ser um problema **NP-completo**, o que significa que ele pertence a NP, pois existe solução que podemos verificar em tempo polinomial e pode ser reduzido para outro problema **NP** em tempo polinomial por conta disso, fazendo com que possa sofrer uma transformação polinomial para outro problema que queiramos realizar a prova de **NP e NP-Completude**.

2.3.Prova de NP

Para provar que o **Problema da Partição de Números (Two-Way Number Partitioning Problem – TWNPP)** pertence a **NP**, precisamos mostrar que uma máquina de turing não-determinística consiga resolvê-lo em tempo polinomial e que uma máquina de turing determinística consiga verificar uma solução em tempo polinomial.

Dado uma solução candidata, podemos verificar se ela é válida ou não em tempo polinomial. Em outras palavras, precisamos de uma verificação eficiente (em tempo polinomial) para saber se uma dada divisão do conjunto de números em dois subconjuntos tem somas iguais.

Considerando a definição anterior do **Problema da Partição de Números** para que um problema seja **NP**, devemos ser capazes de verificar se uma solução é válida em tempo polinomial. Isso significa que, dado uma divisão candidata do conjunto **S** em dois subconjuntos **S1** e **S2**, devemos ser capazes de verificar se as somas dos elementos de **S1** e **S2** são iguais.

2.3.1-Possível entrada da solução:

Consideremos um conjunto **S={S1,S2,…,Sn}** e uma solução candidata que consiste em uma divisão de S em dois subconjuntos **S1** e **S2**).

2.3.2-Verificação das somatórias:

Verifique se a soma dos elementos de S1 é igual à soma dos elementos de **S2**

$$\sum_{x \in S_1} x = \sum_{x \in S_2} x$$

2.3.3-Decisão:

-Se a soma de **S1** for igual à soma de **S2**, então a solução candidata é válida.

-Caso contrário, a divisão candidata não é uma solução válida.

2.3.4-Tempo de verificação:

-Tempo necessário para verificar se as somas de **S1** e **S2** são iguais é O(n), onde n é o número de elementos no conjunto **S**, pois precisamos somar os elementos de **S1** e **S2** uma vez.

-A verificação pode ser feita em tempo linear, o que é claramente polinomial em relação ao tamanho da entrada **S**.

Portanto, dado uma divisão candidata **S1** e **S2** do conjunto **S**, podemos verificar se as somas dos elementos desses subconjuntos são iguais em tempo **O(n)**. Como este processo pode ser realizado em tempo polinomial, podemos concluir que o **Problema da Partição de Números** pertence à classe **NP**.

2.4.Prova de NP-Completude

A prova de NP-completude do P**roblema da Partição de Números (Two-Way Number Partitioning Problem, TWNPP)** é feita por redução polinomial a partir do **Problema da Soma de Subconjuntos (Subset Sum Problem)**, que já é conhecido como **NP-completo**. O objetivo é mostrar que, se conseguirmos resolver o **Problema da Partição de Números** de forma eficiente, então também podemos resolver o **Problema da Soma de Subconjuntos** de forma eficiente, o que implica que **o Problema da Partição de Números é NP-completo.**

Considerando as definições de cada problema já abordado anteriormente, para provar que o **Problema da Partição de Números** é **NP-completo**, faremos duas coisas:

Provar que o **Problema da Partição de Números** está em NP, ou seja, podemos verificar uma solução candidata em tempo polinomial(já feito).

Reduzir um problema conhecido como NP-completo (**Problema da Soma de Subconjuntos**) para o **Problema da Partição de Números** em tempo polinomial. Isso implica que, se conseguirmos resolver o Problema da Partição de Números eficientemente, então também poderemos resolver o **Problema da Soma de Subconjuntos** eficientemente, o que provará que o **Problema da Partição de Números é NP-completo.**

2.4.1-Primeiro passo: Provar que o Problema da Partição de Números está em NP

Mostramos anteriormente que o **Problema da Partição de Números está em NP**, pois, dado uma partição candidata (**S1**,**S2**), podemos verificar em tempo polinomial se a soma de **S1** é igual à soma de **S2**.Portanto, o **Problema da Partição de Números está em NP.**

2.4.2 Segundo passo: Reduzir o Problema da Soma de Subconjuntos para o Problema da Partição de Números

Vamos realizar a redução polinomial do **Problema da Soma de Subconjuntos** para o **Problema da Partição de Números**. O objetivo é mostrar que, dado uma instância do **Problema da Soma de Subconjuntos**, podemos transformá-la em uma instância equivalente do Problema da Partição de Números em tempo polinomial.

2.4.3.Entrada do Problema da Soma de Subconjuntos:

Temos um conjunto S={s1,s2,...,sn} e um número alvo T. Primeiro, vamos calcular a soma total dos elementos de S:

$$S_{\text{total}} = \sum_{i=1}^{n} s_i$$

Se Stotal for ímpar, imediatamente podemos responder não para a instância do **Problema da Partição de Números**, porque não podemos dividir um número ímpar igualmente entre dois subconjuntos de números inteiros. Portanto, a partição será impossível.

Caso contrário, se Stotal for par, definimos um novo alvo T′=Stotal / 2.

2.4.4-Transformação de Instância

Agora, vamos criar uma instância do **Problema da Partição de Números** com o mesmo conjunto S de entrada.

A pergunta será: é possível dividir o conjunto **S** em dois subconjuntos cujas somas sejam iguais?

Se houver um subconjunto S′⊆S tal que a soma de S′ seja T (onde T=T′=(Stotal / 2), então o complemento de S′ em S terá soma também igual a T, pois Stotal =2T.

Isso implica que a partição de S em dois subconjuntos de soma igual é possível.

Portanto, se podemos resolver o **Problema da Partição de Números** para a instância gerada, podemos resolver o **Problema da Soma de Subconjuntos**. Ou seja, a solução do Problema da Partição de Números resolve o Problema da Soma de Subconjuntos.

2.4.5-Conclusão sobre a abordagem:

A redução do **Problema da Soma de Subconjuntos** para o **Problema da Partição de Números** pode ser feita em tempo polinomial, pois a soma de S e o cálculo de T′ podem ser feitos em tempo linear, e as verificações subsequentes também são lineares em relação ao número de elementos de S.

Como o **Problema da Soma de Subconjuntos** é NP-completo, isso implica que o **Problema da Partição de Números** também é NP-completo, pois mostramos que, se conseguirmos resolver o Problema da Partição de Números eficientemente, também poderemos resolver o Problema da Soma de Subconjuntos eficientemente.

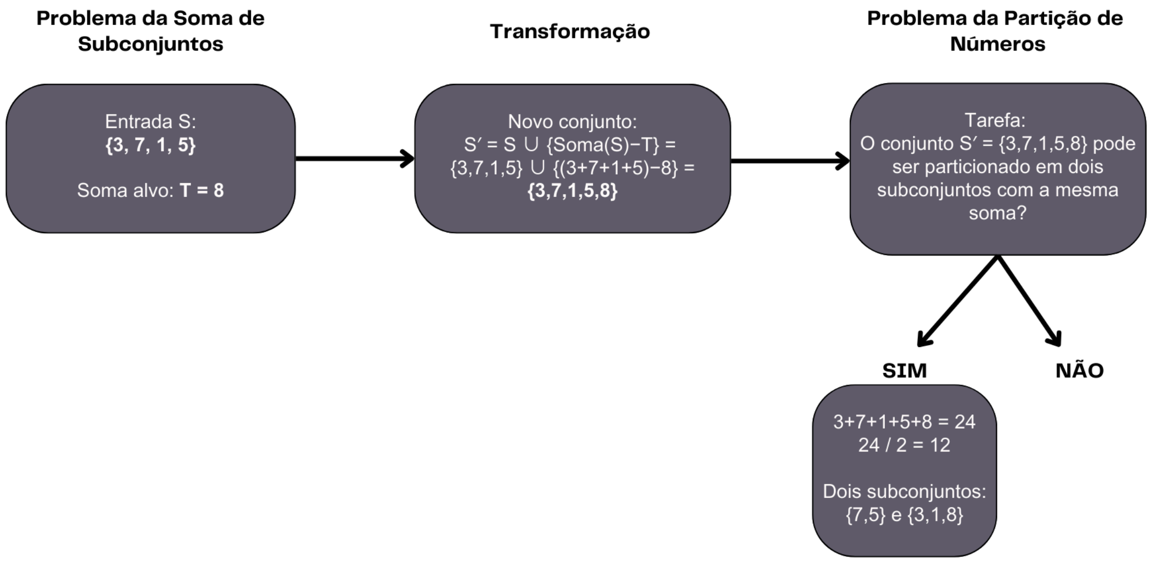
3. Resultados

Nesta seção, apresentamos os resultados obtidos ao utilizar o Problema da Soma de Subconjuntos para provar que o Problema de Partição de Números é NP-Completo.

3.1 Confirmação da Classificação NP - Completa

A partir da redução polinomial do Problema da Soma de Subconjuntos para o Problema de Partição, demonstramos que se houvesse algum algoritmo eficiente para resolver a Partição de Números, então também haveria um para resolver a Soma de Subconjuntos, assim confirmando sua classificação como NP-Completo.

Figura 1. Redução do Problema da Soma de Subconjuntos para o Problema da Partição



Transformação de uma instância do Problema da Soma de Subconjuntos em uma instância equivalente do Problema da Partição.

Portanto, a resposta para essa instância do **Problema da Soma de Subconjuntos** seria **"Sim"**, já que conseguimos encontrar um subconjunto cuja soma é igual a T=8, e a transformação leva a uma solução válida no **Problema da Partição**, encontrando dois subconjuntos onde a soma é igual o valor esperado.

3.2 Comparação de Abordagens

A Tabela 1 apresenta uma comparação entre os tempos de execução da resolução exata e heurística para diferentes instâncias do Problema da Partição de Números.

Tabela 1. Comparação de Métodos de Resolução

Algoritmo	Instância Pequena	Instância Média	Instância Grande
Força Bruta O(2^(n))	0.00000083 s	0.00000079 s	0.00000269 s
Programação Dinâmica O(nS)	0.00025878 s	0.00000076 s	0.00181649 s
Heurística Aproximada O(nlogn)	0.00000291 s	0.00000391 s	0.00000470 s

Aqui estão os tempos médios após 10 execuções para cada instância de pior caso. Para garantir os valores da tabela, é preciso considerar as métricas de medição, as características das instâncias e a implementação do algoritmo. Dessa forma, utilizamos os valores de instância, pequena (20 números), média (30 números) e grande (40 números). Em sua implementação, para força bruta, serão testadas todas as possíveis partições, para a programação dinâmica, utilizamos uma array de soma parcial para decidir se uma partição é possível ou não e para a heurística, um método guloso, ordenando os números e os atribuindo alternadamente aos subconjuntos. Ao final, cada método será executado várias vezes e tirado a média dessas medições, que resultará em nosso tempo de execução.

3.3 Crescimento da Complexidade

Figura 2

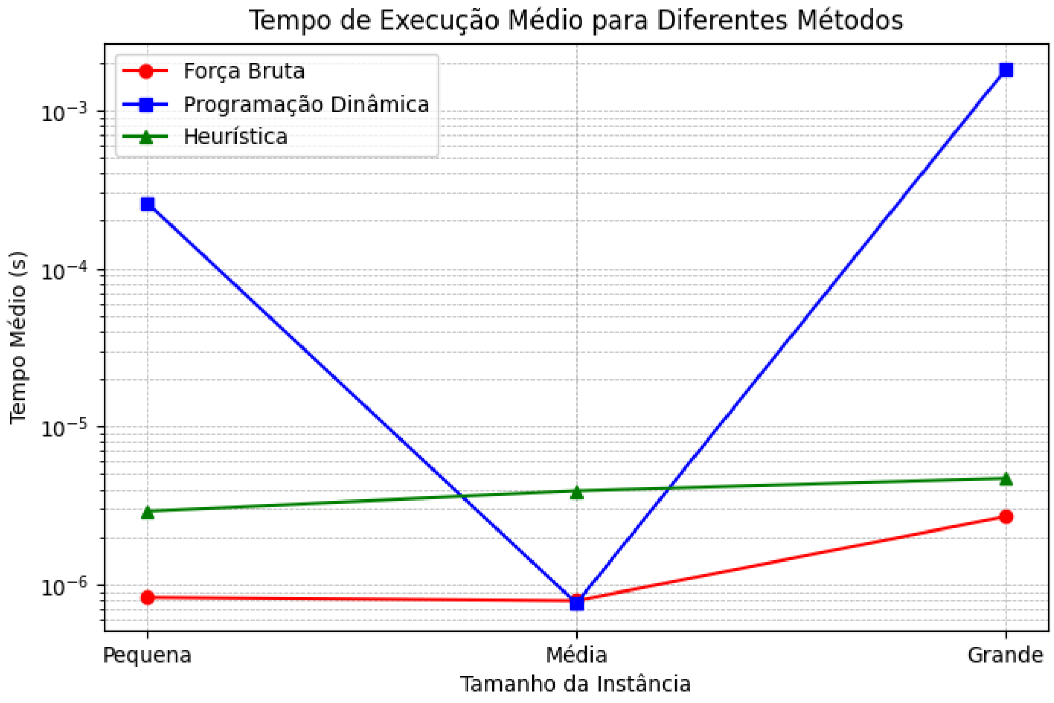


Figura 2, gráfico que representa a Tabela 1 - média das instâncias no pior caso.

4. Conclusões

A resolução do **Problema da Soma de Subconjuntos** é equivalente à do **Problema da Partição de Números**, pois ambos envolvem verificar a existência de subconjuntos com somas específicas. No Problema da Soma de Subconjuntos, buscamos um subconjunto cuja soma seja igual a um valor dado T, enquanto no Problema da Partição, procuramos dividir um conjunto em dois subconjuntos com somas iguais. A redução de um problema para o outro, caso T seja a soma de todos os elementos do conjunto dividida por dois, mostra que, se pudermos resolver um seremos capazes de resolver o outro, já que encontrar uma solução para o Problema da Partição implica em encontrar um subconjunto com a soma desejada no Problema da Soma de Subconjuntos. A partir dos resultados obtidos com o exemplo anterior o conjunto S´ = {3, 7, 1, 5, 8} pode ser dividido em dois subconjuntos com a mesma soma, {7, 5} , {3, 1 , 8}. Logo, é possível fazer a redução de um problema para o outro, uma vez que a soma dos subconjuntos de S´ foram igual a T/2, o que atende aos dois problemas.

Pode-se concluir que, como o **Problema da Partição de Números** tem a sua solução verificada em tempo polinomial, e o **Problema da Soma de Subconjuntos** (já conhecido por ser NP-Completo) foi reduzido a ele como no exemplo anterior, isso prova que o problema inicial também é NP-Completo, que, além de ter sua solução em tempo polinomial para qualquer tamanho de entrada, a equivalência dos dois problemas prova a complexidade intrínseca de ambos visto que a solução eficiente de um resulta na solução eficiente do outro.

5. Referências

Boulic, R. and Renault, O. (1991) "3D Hierarchies for Animation", In: New Trends in Animation and Visualization, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons ltd., England.

Dyer, S., Martin, J. and Zulauf, J. (1995) "Motion Capture White Paper", http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html, December.

Holton, M. and Alexander, S. (1995) "Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials", Computer Graphics: Developments in Virtual Environments, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd., p. 449-460.

Knuth, D. E. (1984), The TeXbook, Addison Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In *Advances in Computer Science*, pages 555–566. Publishing Press.

Subset sum. 02 nov. 2020. Disponível em: https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/mochila-su bsetsum.html. Acesso em: 27 abr. 2022.

Problema da Mochila Multicritério. Maio. 2018. Disponível em: https://fenix.tecnico.ulisboa.pt/downloadFile/395137622440/Disserta %C3%A7%C3%A3o.pdf. Acesso em: 26 abr. 2022.

Knapsack Problem. 19 abr. 2022. Disponível em: https://mathworld.wolfram.com/KnapsackProblem.html. Acesso em: 26 abr. 2022.

The Knapsack Problem. 24 jan. 2003. Disponível em: https://personal.utdallas.edu/~scniu/OPRE-6201/documents/DP3-Kn apsack.pdf. Acesso em: 27 abr. 2022.

Karmarkar, Narenda; Karp, Richard M (1982), «The Differencing Method of Set Partitioning», University of California at Berkeley: Computer Science Division (EECS), Technical Report UCB/CSD 82/113

Karp, R. M. (1972). Reducibility Among Combinatorial Problems. Em R. E. Miller & J. W. Thatcher (Eds.), Complexity of Computer Computations (pp. 85-103). New York: Plenum Press.