Contagem de partições por meio de suas representações matriciais

Palavras-Chave: Partições, Combinatória, Matrizes de duas linhas

Autores:
THIAGO FELIPE CASTRO CARRENHO - UNICAMP
Prof. Dr. JOSÉ PLÍNIO DE OLIVEIRA SANTOS - UNICAMP

Introdução

Uma **partição irrestrita** Ω é uma coleção de inteiros positivos $\{c_1, c_2, \cdots, c_s\}$ cuja soma é $n \in \mathbb{N}$, coleção esta que escolhemos pôr em ordem decrescente, $c_1 \geq c_2 \geq \cdots \geq c_s$. Além disso, uma **partição restrita** deve obedecer às restrições a que está sujeita. Neste projeto, faremos uso de duas restrições:

- **Parte mínima**: A menor das partes da partição deve ser no mínimo $c \in \mathbb{N}$, $c < n - 1$, isto é, $c_s \geq 1$;
- **Diferença mínima entre partes**: Devido à ordenação decrescente das partes, restringe-se que uma parte seja no mínimo a parte seguinte mais um $\lambda \in \mathbb{N} \cup \{0\}$, isto é, $c_i \geq c_{i+1} + \lambda$, $\forall\ i = 1, 2, \cdots, c_{s-1}$.

Dessa maneira, tem-se as partições irrestritas como um caso específico, em que $c = 1$ e $\lambda = 0$, no qual as restrições se tornam redundantes.

Assim, define-se o conjunto de partições $\mathbb{P}(n, c, \lambda)$ como o conjunto de partições de n com parte mínima no mínimo c e diferença mínima entre partes λ, com $p(n) = \#\mathbb{P}(n, 1, 0)$ o número de partições irrestritas de n.

Seguinte a este estabelece-se o conjunto $\mathbb{M}(n, c, \lambda)$, introduzido em (GODINHO; SANTOS, 2020) como o conjunto de todas matrizes de duas linhas do tipo

$$M = \begin{pmatrix} a_1 & a_2 & \cdots & a_s \\ b_1 & b_2 & \cdots & b_s \end{pmatrix}$$

onde $a_j, b_j \in \mathbb{N} \cup \{0\}$ e

$$a_s = c,\ a_j = a_{j+1} + b_{j+1} + \lambda \text{ e } \sum_{i=1}^{s}(a_i + b_i) = n$$

Constrói-se, então, a bijeção entre $\mathbb{P}(n, c, \lambda)$ e $\mathbb{M}(n, c, \lambda)$, em que $M \in \mathbb{M}(n, c, \lambda)$ é a representação matricial de uma única partição $\Omega \in \mathbb{P}(n, c, \lambda)$, de forma que a matriz

$$M = \begin{pmatrix} a_1 & a_2 & \cdots & a_s \\ b_1 & b_2 & \cdots & b_s \end{pmatrix}$$

representa a partição $\Omega = \{a_1 + b_1, a_2 + b_2, \cdots, a_s + b_s\}$.

As restrições impostas sobre a matriz fazem com que as restrições da partição restrita sejam obedecidas. Portanto, para restrições e casos distintos, as matrizes de duas linhas são definidas de maneira diferente, como ocorre em (SANTOS; MATTE, 2018), que para satisfazer as necessidades da pesquisa, a matriz foi definida de outra forma.

Na tabela 1 abaixo tem-se um exemplo de todas as partições irrestritas de 5, isto é, com parte mínima $c = 1$ e diferença mínima entre partes $\lambda = 0$, e suas respectivas representações matriciais, ou seja o número de partições irrestritas de 5 é $p(5) = \#\mathbb{P}(5,1,0) = \#\mathbb{M}(5,1,0) = 7$.

Na tabela 2 abaixo tem-se um exemplo de todas as partições restritas de 9 com parte mínima $c = 2$ e diferença mínima entre partes $\lambda = 1$, e suas respectivas representações matriciais, ou seja o número de partições nesse caso é $\#\mathbb{P}(9,2,1) = \#\mathbb{M}(9,2,1) = 5$.

$c_1 + c_2 + \cdots + c_s$	$\{c_1, c_2, \cdots, c_s\}$	$\begin{pmatrix} a_1 & a_2 & \cdots & a_s \\ b_1 & b_2 & \cdots & b_s \end{pmatrix}$
5	$\{5\}$	$\begin{pmatrix} 1 \\ 4 \end{pmatrix}$
$4+1$	$\{4,1\}$	$\begin{pmatrix} 1 & 1 \\ 3 & 0 \end{pmatrix}$
$3+2$	$\{3,2\}$	$\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$
$3+1+1$	$\{3,1,1\}$	$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 0 & 0 \end{pmatrix}$
$2+2+1$	$\{2,2,1\}$	$\begin{pmatrix} 2 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
$2+1+1+1$	$\{2,1,1,1\}$	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$
$1+1+1+1+1$	$\{1,1,1,1,1\}$	$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

Tabela 1: $\mathbb{P}(5,1,0)$ e $\mathbb{M}(5,1,0)$

$c_1 + c_2 + \cdots + c_s$	$\{c_1, c_2, \cdots, c_s\}$	$\begin{pmatrix} a_1 & a_2 & \cdots & a_s \\ b_1 & b_2 & \cdots & b_s \end{pmatrix}$
9	$\{9\}$	$\begin{pmatrix} 2 \\ 7 \end{pmatrix}$
$7+2$	$\{7,2\}$	$\begin{pmatrix} 3 & 2 \\ 4 & 0 \end{pmatrix}$
$6+3$	$\{6,3\}$	$\begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}$
$5+4$	$\{5,4\}$	$\begin{pmatrix} 5 & 2 \\ 0 & 2 \end{pmatrix}$
$4+3+2$	$\{4,3,2\}$	$\begin{pmatrix} 4 & 3 & 2 \\ 0 & 0 & 0 \end{pmatrix}$

Tabela 2: $\mathbb{P}(9,2,1)$ e $\mathbb{M}(9,2,1)$

Partindo disto, organiza-se as matrizes numa árvore e conta-se quantos nós existem por geração na árvore. Neste momento a pesquisa foi subdividida, numa primeira parte, a dedução do algoritmo, conforme a análise feita no artigo (GODINHO; SANTOS, 2020), apenas para o caso irrestrito, em seguida, na segunda parte, foi estudado mais a fundo a organização em árvore, com a adição de funções e a explicação da montagem também para o caso restrito.

Desenvolvimento da árvore de matrizes

Sabendo que cada matriz da maneira definida representa uma partição, organizou-se as matrizes de acordo com o número de zeros no final da segunda linha, todas as matrizes com t_1 zeros no final da segunda linha são descendentes de uma mesma matriz raiz, o bloco $B(t_1)$.

Os blocos constituem a primeira geração da árvore e são definidos como:

$$B(0) = \begin{pmatrix} c \\ m(0) \end{pmatrix}, B(1) = \begin{pmatrix} c+\lambda & c \\ m(1) & 0 \end{pmatrix}, \cdots, B(r) = \begin{pmatrix} c+r\lambda & \cdots & c+\lambda & c \\ m(r) & \cdots & 0 & 0 \end{pmatrix}.$$

onde $m(t_1) = n - \left(c(t_1+1) + \lambda \dfrac{t_1(t_1+1)}{2} \right)$ representa o quanto falta para a soma de todos os elementos da matriz ser n, isto é, deve ser um inteiro não negativo.

Para os descendentes, precisamos definir

$$A(t_1, t_2) = \begin{pmatrix} c+t_1\lambda & c+(t_1-1)\lambda & \cdots & c+\lambda & c \\ 1+t_2 & 0 & \cdots & 0 & 0 \end{pmatrix} \text{ e } L(x) = c + (x+1)\lambda + 1.$$

Os descendentes dos blocos são as matrizes $F(t_1, t_2)$ definidos por

$$F(t_1, t_2) = \begin{pmatrix} L(t_1) + t_2 \\ m(t_1, t_2) \end{pmatrix} \uplus A(t_1, t_2)$$

para $t_2 = 0, 1, \cdots, \lfloor \frac{m(t_1, 0)}{2} \rfloor$, sendo $m(t_1, t_2) = m(t_1) - 2t_2 - L(t_1) - 1$.

A cada geração a seguir, adiciona-se uma nova variável (o que explica a contagem ser feita por linha), e de uma geração para outra, adiciona-se uma coluna a esquerda, isto é, na ℓ-ésima geração, as matrizes dependerão de $t_1,\ t_2, \cdots t_\ell$, e em cada uma delas terá uma função $m(t_1, \cdots, t_\ell)$, que enquanto devolve um valor inteiro não negativo, complementa a matriz de forma que some n, quando devolve um valor negativo, mostra que não existe matriz com esses valores para as variáveis, e, quanto à construção da matriz, onde estava $m(t_1, \cdots, m(t_{\ell-1})$ na matriz pai, se coloca t_ℓ, adiciona-se uma coluna à esquerda, em que o elemento da linha superior, a_1, já é definido, e o elemento inferior, b_1 se torna $m(t_1, \cdots, t_\ell)$.

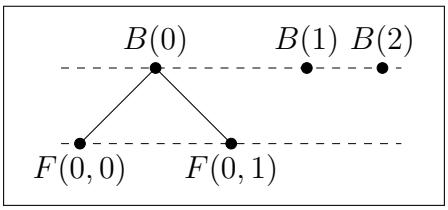
No exemplo da Tabela 2, os blocos são

$$B(0) = \begin{pmatrix} 2 \\ 7 \end{pmatrix}, B(1) = \begin{pmatrix} 3 & 2 \\ 4 & 0 \end{pmatrix}, B(2) = \begin{pmatrix} 4 & 3 & 2 \\ 0 & 0 & 0 \end{pmatrix}$$

e os primeiros descendentes são

$$F(0,0) = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}, F(0,1) = \begin{pmatrix} 5 & 2 \\ 0 & 2 \end{pmatrix},$$

que já constituem todas as matrizes da árvore, fazendo com que ela seja conforme abaixo.



Por fim, basta contar quais são os valores de $t_1, \cdots, t_\ell$ para os quais $m(t_1, \cdots, t_\ell)$ devolve um valor inteiro não negativo, que se tem quantos nós existem na ℓ-ésima geração da árvore, sem precisar saber quais são as matrizes, ou mesmo qual partição elas representam.

Desenvolvimento do algoritmo do caso irrestrito

Definem-se o seguinte polinômio, sua raiz e a constante ℓ_0 como

$$\mathcal{P}_0(x) = n - 2(x+1),\ x_0 = \frac{n}{2} - 1,\ \ell_0 = \left\lfloor \frac{n-2}{2} \right\rfloor + 1 = \left\lfloor \frac{n}{2} \right\rfloor,$$

e a partir deste definem-se os próximos polinômios e suas respectivas raízes por

$$\mathcal{P}_1(x) = n - 1 - x \text{ e } \mathcal{P}_j(x) = n - 2j - x \text{ para } j = 2, \cdots, \ell_0,$$
$$r_1 = n - 1 \text{ e } r_j = n - 2j \text{ para } j = 2, \cdots, \ell_0.$$

Ao tomar a visão de árvore, tem-se que ℓ_0 representa o número de gerações na árvore, e o número de elementos na primeira geração é $\mathbb{D}_1 = r_1 + 1 = n$, e nas gerações $\ell = 2, \cdots, \ell_0$ é $\mathbb{D}_\ell$ definido por:

$$\mathbb{D}_\ell = \sum_{t_1=0}^{r_\ell} \sum_{t_2=0}^{T_2^{(\ell)}} \cdots \sum_{t_{\ell-1}=0}^{T_{\ell-1}^{(\ell)}} \left(\left\lfloor \frac{\mathcal{P}_\ell(t_1) - \sum_{i=3}^{\ell} i \cdot t_{\ell-i+2}}{2} \right\rfloor + 1 \right)$$

onde

$$T_j^{(\ell)} = T_j^{(\ell)}(t_1, t_2, \cdots, t_{j-1}) = \left\lfloor \frac{\mathcal{P}_\ell(t_1) - \sum_{i=\ell-j+3}^{\ell} i \cdot t_{\ell-i+2}}{\ell - (j-2)} \right\rfloor.$$

Por fim, obtido todo $\mathbb{D}_\ell$ para $\ell = 1, \cdots, \ell_0$, tem-se o número de matrizes em cada geração, logo, o total de matrizes na família é, simplesmente, a soma desses valores:

$$p(n) = \sum_{\ell=1}^{\ell_0} \mathbb{D}_\ell = (r_1 + 1) + \sum_{\ell=2}^{\ell_0} \mathbb{D}_\ell$$

A dedução destas fórmulas está em (GODINHO; SANTOS, 2020). A implementação dos somatórios aninhados foi feito recursivamente, e a implementação pode ser encontrada em (CARRENHO, 2021).

Resultados e Discussão

O principal resultado é o próprio algoritmo mostrado acima, implementado em funções em MATLAB. Fazendo um laço em n dessa função, conseguimos analisar melhor o comportamento do algoritmo.

Além do valor de $p(n)$, coletamos o tempo de processamento do algoritmo para os valores calculados, que foram de $n = 1$ até $n = 167$, e estão presentes no arquivo `listadeparticoes.txt`, disponível em (CARRENHO, 2021), e uma separação por intervalos na tabela 3 abaixo.

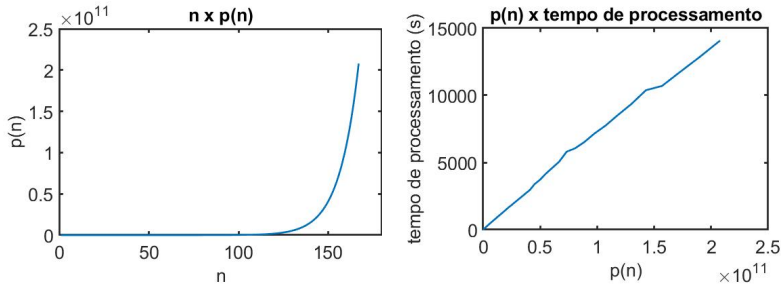
Tempo de Processamento	Valores de n
Até 1 segundo	$n = 1 : 76$
De 1 segundo a 1 minuto	$n = 77 : 111$
De 1 minuto até 1 hora	$n = 112 : 151$
Mais de 1 hora	$n = 152 : 167$

Tabela 3: Intervalos de Tempo de Processamento

O gráfico de $n \times p(n)$, Figura 1(a) abaixo, comprova que a função $p(n)$ realmente explode para n crescente, e o gráfico de $p(n)\times$tempo de processamento, Figura 1(b) abaixo, mostra um comportamento quase linear entre essas duas variáveis, assim, conforme n cresce, o tempo de processamento também explode.

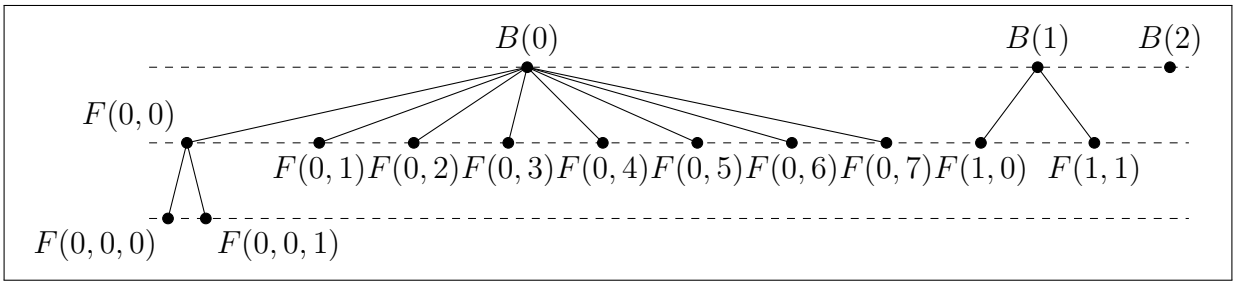
Para o caso restrito, que engloba o irrestrito, o resultado principal é a possibilidade de calcular o número de partições de acordo com as duas restrições sem precisar colocar por extenso cada partição ou cada matriz, apenas analisando as funções, especialmente $m(t_1, \cdots, t_\ell)$, e montando a árvore, contando quantos nós ela possui a cada geração.

Por exemplo, vê-se que $\#\mathbb{M}(32, 7, 2) = 15$ pela árvore abaixo.



(a) $n \times p(n)$ (b) $p(n)\times$tempo

Figura 1: Gráficos



Conclusão

Conclui-se que o algoritmo calcula o número de partições irrestritas de n, como era seu objetivo, confirmando o crescimento cada vez maior da função $p(n)$.

Além disso, é possível concluir que o algoritmo é eficiente e viável, pois calcula valores até $p(111)$ em menos de um minuto. Entretanto, o algoritmo perde valor para grandes valores de n, já que o tempo se torna impraticável, isto é, o algoritmo tem um 'limite', que depende de quanto tempo o computador pode estar alocado apenas para cálculo desta função.

Já para o caso restrito, a representação matricial mostrou uma nova maneira de calcular o número de partições para as restrições propostas, e isso abre portas para outras aplicações para a representação. Como uma continuação deste projeto, é possível implementar o caso restrito, assim como incluir outras restrições para as partições.

Referências

CARRENHO, Thiago F. C. **IC-Partições**. [S. l.: s. n.], 2021. Disponível em: ¡https://github.com/ThiagoCarrenho/IC-particoes.git¿.

GODINHO, Hemar; SANTOS, José Plínio O. A family of partitions equinumerous with the set of nodes of a family of trees. **INTEGERS**, v. 20, 2020. Disponível em: ¡http://math.colgate.edu/~integers/u101/u101.pdf¿.

SANTOS, José Plínio O.; MATTE, Marilia L. A New Approach to Integer Partition. **Bulletin of the Brazilian Mathematical Society**, 2018. DOI: https://doi.org/10.1007/s00574-018-0082-z.

SANTOS, José Plínio O.; MELLO, Margarida P.; MURARI, Idani T.C. **Introdução à Análise Combinatória**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2007.

SANTOS, José Plínio O.; MONDEK, Paulo; RIBEIRO, Andréia C. New Two-Line Arrays Representing Partitions. **Annals of Combinatorics**, v. 15, n. 341, 2011. DOI: https://doi.org/10.1007/s00026-011-0099-0.