Uma nova perspectiva para o Algoritmo de Euclides: Uma abordagem computacional para o cálculo dos coeficientes da Relação de Bézout

A new perspective to the Euclidean Algorithm: A computational approach for calculating the coefficients of the Bézout's Identity

Ana Carla Quallio Rosa[1], Fernando Cézar Gonçalves Manso[2], Wellington José Corrêa[3]

RESUMO

A Teoria dos Números é uma disciplina matemática que se dedica ao estudo das propriedades e relações dos números inteiros. Dentre os conceitos fundamentais neste campo, destaca-se o Máximo Divisor Comum (mdc). Uma abordagem eficaz para calcular o mdc de maneira computacional reside no Algoritmo de Euclides, que recorre à recursão para progressivamente simplificar o problema até que este se reduza a uma questão trivial. A versão estendida do Algoritmo de Euclides possibilita estabelecer uma relação entre o Máximo Divisor Comum de dois números inteiros e as suas respectivas combinações lineares, por intermédio da Relação de Bézout. O escopo deste estudo se concentra em apresentar uma nova abordagem para o Algoritmo Estendido de Euclides, determinando os coeficientes que compõem o Máximo Divisor Comum mediante um conjunto de iterações. Desse modo, o método desenvolvido é aplicado de forma computacional para verificar a precisão das soluções obtidas. Essa avaliação abarca desde situações elementares da função mdc até cenários que, adotando uma abordagem convencional, demandam muitas operações aritméticas. Por meio da análise assintótica, foi possível concluir que o novo algoritmo possui o mesmo custo da versão tradicional, isto é, $O(log(min(a,b)))$.
PALAVRAS-CHAVE: Algoritmo de Euclides; Implementação computacional; Teoria dos Números.

ABSTRACT

Number Theory is a mathematical discipline dedicated to the study of the properties and relationships of integers. Among the fundamental concepts in this field, the Greatest Common Divisor (GCD) stands out. An effective approach to computationally calculate the GDC resides in the Euclidean Algorithm, which employs recursion to progressively simplify the problem until it reduces to a trivial matter. The extended version of the Euclidean Algorithm allows establishing a relationship between the Greatest Common Divisor of two integers and their respective linear combinations, through Bézout's Identify. The scope of this study focuses on presenting a new approach to the Extended Euclidean Algorithm, determining the coefficients that compose the Greatest Common Divisor through a set of iterations. Thus, the developed method is computationally applied to verify the accuracy of the obtained solutions. This evaluation encompasses elementary cases of the GCD function to scenarios that, following a conventional approach, require many arithmetic operations. Through asymptotic analysis, it was possible to conclude that the new algorithm has the same complexity as the traditional version, $O(log(min(a, b)))$.
KEYWORDS: Euclidean Algorithm; Computational implementation; Number Theory.

INTRODUÇÃO

A Teoria dos Números é um ramo da Matemática que visa, primordialmente, entender as propriedades e relações entre os números. Segundo Santos (2014), este ramo se desdobra em três principais abordagens: a Teoria Algébrica, voltada ao estudo da álgebra e dos números complexos; a Teoria Analítica, referente à análise real e

[1] Bolsista da Fundação Araucária. Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: anacarlarosa@alunos.utfpr.edu.br. ID Lattes: 3888049909661814.
[2] Docente no Departamento Acadêmico de Química. Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: fmanso@utfpr.edu.br. ID Lattes: 2920136847904900.
[3] Docente no Departamento Acadêmico de Matemática. Universidade Tecnológica Federal do Paraná, Campo Mourão, Paraná, Brasil. E-mail: wcorrea@professores.utfpr.edu.br. ID Lattes: 1045931096324971.

complexa, e a Teoria Elementar, que consiste em técnicas para a validação e comprovação de conceitos inerentes dos números inteiros.

Dentro da Teoria Elementar, há o estudo das propriedades dos divisores comuns, com apresentação do Algoritmo de Euclides. Em suma, esse algoritmo é um método de divisões sucessivas que possibilita encontrar o Máximo Divisor Comum (mdc) entre dois inteiros. Outra variante desse algoritmo, denominada Algoritmo Estendido de Euclides, explora a Relação de Bézout, que postula que o mdc entre dois números pode ser expresso como a combinação linear entre esses mesmos números. Essa relação possui uma significativa relevância no campo da Computação, particularmente no contexto do processo de cifragem e decifragem do sistema de criptografia RSA (STALLINGS; VIEIRA, 2008).

Este trabalho tem como objetivo apresentar uma nova perspectiva para o Algoritmo Estendido de Euclides, de modo a determinar os coeficientes da Relação de Bézout em um conjunto de iterações. Desse modo, implementou-se computacionalmente o método desenvolvido, na linguagem de programação *Python*, a fim de verificar a corretude dos cálculos diante de diferentes possibilidades para a função mdc.

REFERENCIAL TEÓRICO

A concepção de divisibilidade entre inteiros é de suma importância para a Teoria dos Números. Por definição, sejam dois inteiros *a* e *b*, com $a \neq 0$, dizemos que *a* divide *b*, escrevendo como $a \mid b$, se e somente se *a* é um divisor ou fator de *b*, ou ainda, *b* é um múltiplo de *a* (CORMEN *et al.*, 2012). Com base nessa noção, podemos introduzir o conceito de divisor comum. Um inteiro *d* é considerado um divisor comum de a e de *b* se e somente se *d* é divisor de *a* e de $b - a$. Consequentemente, o máximo divisor comum é identificado como o maior dos divisores comuns de *a* e *b*, sendo denotado por *mdc(a,b)*.

Com o intuito de demonstrar a existência do Máximo Divisor Comum, Euclides apresentou o Teorema da Recursão, o qual estabelece que, dados a e $b \in \mathbb{N}$,

$$mdc(a,b) = mdc(b,\ a \bmod b) \tag{1}$$

Tal teorema proporciona a diminuição da complexidade da fatoração dos números por meio da recursividade, até torná-la trivial. Além disso, temos a Relação de Bézout, a qual afirma que dados dois inteiros *a* e *b*, sendo ambos não nulos, o menor elemento do conjunto:

$$a\mathbb{Z} + b\mathbb{Z} = \{a \times m + b \times n,\ m\ e\ n \in \mathbb{Z}\}\ \text{é o}\ mdc(a,b) \tag{2}$$

Em suma, esta relação estabelece que o máximo divisor comum entre dois inteiros *a* e *b* pode ser escrito como a combinação linear entre *a* e *b* (SANTOS, 2014). Os coeficientes dessa relação são obtidos pelo Algoritmo Estendido de Euclides. Para ilustrar o funcionamento deste algoritmo, tomemos o exemplo do cálculo de *mdc(372,162)*. A primeira etapa consiste em determinar o máximo divisor comum, escrevendo cada resultado em posições de uma tabela. Inicialmente, tem-se que:

$$a = bq_0 + r_0,\ sendo\ b = r_0q_1 + r_1 \tag{3}$$

Generalizando esse processo, obtemos a Figura 1.

Figura 1 – Generalização do algoritmo de Euclides

		q_0	q_1	…	q_{n-1}	q_n	q_{n+1}
a	b	r_0	r_1	…	r_{n-1}	r_n	r_{n+1}

Fonte: Elaborado pelos autores (2023).

Observa-se que a primeira linha da tabela contém os quocientes e a segunda linha apresenta os restos das divisões. O penúltimo elemento, r_n, corresponde ao máximo divisor comum. No caso de *mdc(372,162)*, os resultados são mostrados na Figura 2.

Figura 2 – Algoritmo de Euclides para *mdc(372,162)*

		2	3	2	1	2
372	162	48	18	12	6	0

Fonte: Elaborado pelos autores (2023).

Portanto, temos que *mdc(372,162)* é igual a 6. Nesse contexto, o Algoritmo de Euclides proporciona as seguintes relações:

$$\begin{aligned} 6 &= 18 - 1.12 \\ 12 &= 48 - 2.18 \\ 18 &= 162 - 3.48 \\ 48 &= 372 - 2.162 \end{aligned} \tag{4}$$

O que implica que:

$$\begin{aligned} 6 &= 18 - (48 - 2.18) = 3.18 - 48 = 3(162 - 3.48) - 48 = \\ &= 3.162 - 10.48 = 3.162 - 10(372 - 2.162) = -10.372 + 23.162 \end{aligned} \tag{5}$$

Em outras palavras, ao manipular os coeficientes da Figura 2, podemos expressar *mdc(372,162)* como uma combinação linear, na qual *m = -10* e *n = 23*, de tal forma que:

$$mdc(372, 162) = 6 = -10.372 + 23.162 \tag{6}$$

Esse é o formato tradicional do Algoritmo Estendido de Euclides. Diante dessa perspectiva, pode-se apresentar o método desenvolvido.

MÉTODO DESENVOLVIDO PARA O CÁLCULO DOS COEFICIENTES

O primeiro passo para calcular os coeficientes *x* e *y* no método desenvolvido consiste em determinar o máximo divisor de dois inteiros *a* e *b*, seguindo o procedimento ilustrado na Figura 1. Posteriormente, divide-se os elementos da segunda linha da tabela pelo mdc, isto é, por r_n, como ilustra a Figura 3.

Figura 3 – Elementos do máximo divisor comum no método desenvolvido

		q_0	q_1	…	q_{n-1}	q_n	q_{n+1}
$\frac{a}{r_n}$	$\frac{b}{r_n}$	$\frac{r_0}{r_n}$	$\frac{r_1}{r_n}$	…	$\frac{r_{n-1}}{r_n}$	$\frac{r_n}{r_n}$	0

Fonte: Elaborado pelos autores (2023).

O segundo passo envolve a verificação da paridade da posição *n*, a fim de iniciar um conjunto de iterações para o cálculo de alfa — variável utilizada para determinar os coeficientes *x* e *y*. Caso *n* seja par, obtemos:

$$i = \frac{n}{2} \tag{7}$$

Caso contrário:

$$i = \frac{n-1}{2} \tag{8}$$

Assim, as iterações são iniciadas com:

$$\alpha_0 = 1 \text{ até } \alpha_i = \frac{\alpha_{i-1} \cdot \frac{r_0}{r_n} - q_2}{\frac{r_2}{r_n}} \tag{9}$$

Por fim, os coeficientes *x* e *y* são obtidos por meio das equações:

$$y = \frac{\alpha_i \cdot \frac{a}{r_n} - q_0}{\frac{r_0}{r_n}} \; e \; x = \frac{y \cdot \frac{b}{r_n} - 1}{-\left(\frac{a}{r_n}\right)} \tag{10}$$

Para exemplificar, considere novamente mdc(372,162). Primeiramente, dividimos os elementos da segunda linha por r_n, obtendo a Figura 4.

Figura 4 – Elementos do máximo divisor comum no método desenvolvido

		2	3	2	1	2
62	27	8	3	2	1	0

Fonte: Elaborado pelos autores (2023).

Em seguida, pode-se calcular o número de iterações como:

$$i = \frac{3-1}{2} = 1 \tag{11}$$

Com isso, temos:

$$\alpha_0 = 1 \text{ até } \alpha_1 = \frac{\alpha_{i-1} \cdot \frac{r_0}{r_n} - q_2}{\frac{r_2}{r_n}} = \frac{1 \cdot \frac{8}{1} - 2}{\frac{2}{1}} = 3 \tag{12}$$

Desse modo, determinando os coeficientes, obtemos:

$$y = \frac{\alpha_i \cdot \frac{a}{r_n} - q_0}{\frac{r_0}{r_n}} = \frac{3 \cdot \frac{62}{1} - 2}{\frac{8}{1}} = 23;\ x = \frac{y \cdot \frac{b}{r_n} - 1}{-\left(\frac{a}{r_n}\right)} = \frac{23 \cdot \frac{27}{1} - 1}{-\left(\frac{62}{1}\right)} = -10 \quad (13)$$

DESENVOLVIMENTO DA IMPLEMENTAÇÃO COMPUTACIONAL

O processo de desenvolvimento do algoritmo iniciou com a criação de um pseudocódigo, conforme ilustra a Figura 5, com o objetivo de proporcionar uma estrutura lógica mais clara e concisa para a implementação.

Figura 5 – Pseudocódigo do algoritmo desenvolvido

```
MDC(a, b)
1 i = 0  // θ(1)
2 while resto != 0
3       do resto = a % b
4       resto_i[i] = resto
5       quociente_i[i] = a / b
6       a, b = b, resto
7       i += 1
8 return resto_i, quociente_i
MDC_ESTENDIDO(a, b)
1 resto_i, quociente_i = MDC(a,b)  // O(log(min(a,b)))
2 resto_i, quociente_i = DIVISAO(resto_i, quociente_i) // θ(n)
3 n = len(resto_i) - 1  // θ(1)
4 alfa[0] = 1 // θ(1)
5 if n % 2 == 0
6       i = n / 2
7 else
8       i = (n - 1)/2
9 for j = 1 to i
10      alfa[j] = (alfa[j - 1] * resto_i[0] - quociente_i[2]) / resto_i[2]
11 y = (alfa[i] * a - quociente_i[0] ) / resto_i[0] // θ(1)
12 x = (y * b - 1) / -a // θ(1)
13 return x, y
```

Fonte: Elaborado pelos autores (2023).

Posteriormente, escolheu-se a linguagem de programação *Python* para a implementação, por conta da facilidade de criação e alocação de itens. Nesse sentido, o desenvolvimento teve início com o tratamento de propriedades elementares da função mdc. Em situações não triviais, o método é invocado, retornando os coeficientes *x* e *y* que satisfazem a Relação de Bézout.

Como a demonstração matemática do algoritmo desenvolvido ainda não foi realizada, utilizou-se a implementação como uma forma de validação. Nesse sentido, foram feitos diversos testes, como, por exemplo, mdc(987, 123) e mdc(890, 376), que possuem tamanhos de tabelas distintos, conforme o Algoritmo de Euclides.

No que diz respeito à complexidade, o custo assintótico da versão tradicional é influenciado principalmente pelo cálculo do Máximo Divisor Comum dos inteiros *a* e *b*. Cormen *et al.* (2012) demonstra que mdc(a,b) possui um custo de *O(log(min(a,b)))*, em que *min(a,b)* representa o menor dos dois números.

Em relação ao método desenvolvido, a Figura 5 informa o custo de cada trecho do pseudocódigo. Em algumas linhas, é apresentada a notação assintótica em função de *n*. Ressalta-se que *n* indica o número de posições do vetor que armazena os quocientes ou restos obtidos durante as iterações do Algoritmo de Euclides. Desse modo, *n* também equivale a $O(log(min(a,b)))$, pois depende do cálculo do Máximo Divisor Comum. Logo, apesar de ter outras operações para determinar os coeficientes, assintoticamente o custo do novo método se mantém dominado pela complexidade do Algoritmo de Euclides.

CONSIDERAÇÕES

Este trabalho teve como objetivo apresentar uma nova perspectiva para o Algoritmo Estendido de Euclides, com foco na determinação dos coeficientes da Relação de Bézout em um conjunto de iterações. Em relação à complexidade, nota-se que o custo assintótico do método desenvolvido é o mesmo do tradicional. Desse modo, o algoritmo se mantém eficiente, sendo influenciado pelo cálculo do Máximo Divisor Comum.

A implementação computacional proporcionou resultados que sugerem a eficácia e precisão da abordagem proposta em diversos contextos da função mdc. No entanto, é importante ressaltar que a demonstração matemática do algoritmo desenvolvido requer investigações futuras para consolidar sua validade teórica dentro da Teoria dos Números, de modo a possibilitar novas aplicações em diferentes áreas do conhecimento científico.

Agradecimentos

Agradeço aos meus orientadores, Wellington e Fernando, pelo apoio e contribuições durante a atuação do projeto de pesquisa. O método apresentado foi desenvolvido pelo professor Fernando, que sempre instigou discussões sobre diversos temas da Matemática. Agradeço, também, à Fundação Araucária pelo fomento financeiro.

Disponibilidade de código

A implementação computacional desenvolvida está disponível em um repositório no *GitHub*, por meio do *link*: https://github.com/anacarlaquallio/euclidean-algorithm.

Conflito de interesse

Não há conflito de interesse.

REFERÊNCIAS

CORMEN, Thomas *et al*. **Algoritmos:** teoria e prática. Rio de Janeiro: Elsevier, 2012.

SANTOS, José Plínio de Oliveira. **Introdução à Teoria dos Números**. 3. ed. Rio de Janeiro: IMPA, 2014.

STALLINGS, W.; VIEIRA, D. **Criptografia e segurança de redes:** princípios e práticas. Pearson Prentice Hall, 2008.