Ernane Rosa Martins
(Organizador)

FUNDAMENTOS DA CIÊNCIA DA COMPUTAÇÃO

Atena
Editora
Ano 2019

Ernane Rosa Martins
(Organizador)

Fundamentos da Ciência da Computação

Atena Editora
2019

2019 *by Atena Editora*
Copyright © da Atena Editora
Editora Chefe: Profª Drª Antonella Carvalho de Oliveira
Diagramação e Edição de Arte: Lorena Prestes e Geraldo Alves
Revisão: Os autores

Dados Internacionais de Catalogação na Publicação (CIP)
(eDOC BRASIL, Belo Horizonte/MG)

F981 Fundamentos da ciência da computação / Organizador Ernane Rosa Martins. – Ponta Grossa (PR): Atena Editora, 2019.

Inclui bibliografia
ISBN 978-85-7247-157-2
DOI 10.22533/at.ed.572190703

1. Computação. I. Martins, Ernane Rosa.
CDD 004

Elaborado por Maurício Amormino Júnior – CRB6/2422

APRESENTAÇÃO

A Ciência da Computação estuda as técnicas, metodologias e instrumentos computacionais, visando automatizar os processos e desenvolver soluções com o uso de processamento de dados. Este livro, possibilita conhecer os elementos básicos desta ciência por meio do contato com alguns dos conceitos fundamentais desta área, apresentados nos resultados relevantes dos trabalhos presentes nesta obra, realizados por autores das mais diversas instituições do Brasil.

Assim, são abordando neste livro assuntos importantes, tais como: desenvolvimento de sistema mobile utilizando as plataformas iOS e Android; desenvolvimento de protótipo que trabalha em cenário real de sala de aula e na comparação de algoritmos usados no reconhecimento facial; criação do jogo que explora a criptografia em um ambiente de computação desplugada; construção de simulador que mostra especificamente o comportamento do escalonador First-in First; apresentação de abordagem para orquestração do conhecimento curricular em Ciência da Computação baseado nas matérias do currículo referência para a Ciência da Computação e em estruturas curriculares de cursos de graduação.

Espero que este livro seja útil tanto para os alunos dos cursos superiores de Ciência da Computação quanto para profissionais que atuam nesta importante área do conhecimento. O principal objetivo deste livro é ajudar na fascinante empreitada de compreender a computação perante os mais diferentes desafios do século XXI. Desejo a todos uma excelente leitura e que está obra contribua fortemente com o seu aprendizado.

Ernane Rosa Martins

SUMÁRIO

CAPÍTULO 11

UM ALGORITMO PARA ENCONTRAR UM POLITOPO MAXIMAL DE VÉRTICES EM Z^n INSCRITO EM UMA HIPERESFERA EM R^n

Yuri Tavares dos Passos
CETEC – UFRB
Cruz das Almas – BA – Brazil

Eleazar Gerardo Madriz Lozada
CETEC – UFRB
Cruz das Almas – BA – Brazil

RESUMO: Neste trabalho apresentamos um algoritmo para construir um poliedro de vértices em Z^n, inscrito em uma hiperesfera em R^n tal que entre a hiperesfera e o poliedro não existam elementos de Z^n
PALAVRAS-CHAVE: hiperesfera n-dimensional, poliedro inteiro, algoritmo.

ABSTRACT: In this work we present an algorithm to construct a polyhedron inscribed in a hypersphere in R^n with vertices in Z^n such that between the hypersphere and the polyhedron there are no points in Z^n
KEYWORDS: n-dimensional hypersphere, integer polyhedron, algorithm.

1 I INTRODUÇÃO

Em matemática, um problema de Programação Linear Inteira (PLI) é um modelo de Programação Linear (PL) no qual algumas ou todas as variáveis assumem valores inteiros positivos. Mesmo que pareça “um problema fácil de resolver”, em geral, é um problema NP-difícil. De modo similar ao PL, o PLI tem associado um conjunto de soluções factíveis. Assim, pelo fato de ser um conjunto discreto, as definições e resultados da teoria de poliedros usados para caracterizar o conjunto de soluções factíveis de um PL podem ser usadas para estudar o conjunto de soluções factíveis associado a um PLI, quando é considerado o poliedro gerado pelo fecho convexo (ROCKAFELLAR, 1997) do conjunto de soluções factíveis associada ao PLI. Em geral, um poliedro pode ser decomposto como a soma de um politopo e um cone convexo. Se o poliedro é um conjunto limitado, temos um politopo e é possível obtê-lo a partir das combinações convexas de um subconjunto finito de elementos do poliedro (SCHRIJVER, 1986).

Um fato interessante de ressaltar é que se consideramos um conjunto finito de pontos de coordenadas inteiras no espaço euclideano n-dimensional, o fecho convexo deste conjunto é um politopo. Além disso, pelo fato de ser um conjunto compacto, existe uma família infinita de hiperesferas que contem o conjunto em questão. A partir deste fato, nos fazemos a seguinte pergunta: será que dessas infinitas hiperesferas existiria pelo menos uma que verifique que entre o politopo e a hiperesfera não existam pontos de coordenadas inteiras? Observe que para qualquer hiperesfera de raio maior que um, é imediato que existe um subconjunto finito de pontos com coordenadas inteiras que responda a nossa pergunta, isto é, que entre o fecho convexo e a hiperesfera não tem pontos de coordenadas inteiras.

Esta observação motiva nosso trabalho. Assim, nosso objetivo é apresentar um algoritmo que para uma hiperesfera dada construa o politopo interior a hiperesfera de tal maneira que entre o politopo e a hiperesfera não existam pontos de coordenadas inteiras. De fato, no interior da hiperesfera de centro na origem e raio $\|g\|$, para um $g \in \mathbb{Z}^n$ onde $\|\cdot\|$ denota a norma euclidiana em $\mathbb{R}^n$., existe uma quantidade finita de subconjuntos de $\mathbb{Z}^n$ Se considerarmos o fecho convexo de cada um destes subconjuntos, teremos a família de politopos que estão inscritos na hiperesfera e algum deles deve satisfazer a condição de que entre o politopo e a hiperesfera não existam pontos de coordenadas inteiras.

Em outras palavras, o objetivo central deste trabalho é descobrir qual destes politopos verifica que a interseção entre o interior do conjunto $C_g = \{x \in \mathbb{R}^n; \|x\| \leqslant \|g\|\}$ e o complemento do politopo não tem elementos em $\mathbb{Z}^n$ Para o caso com n = 2, uma ferramenta usando GeoGebra foi apresentada por Maltez e Lozada (MALTEZ; LOZADA, 2015) e (MALTEZ, 2016). O presente trabalho foi publicado anteriormente em (LOZADA et al, 2017).

2 I O PROBLEMA DO POLITOPO MAXIMAL INSCRITO NUMA HIPERESFERA (PMIH)

Seja $g \in \mathbb{Z}^n$ e $C_g = \{x \in \mathbb{R}^n; \|x\| \leqslant \|g\|\}$ onde $\|\cdot\|$ denota a norma euclidiana em $\mathbb{R}^n$. Para um subconjunto *V* não vazio e finito de $\mathbb{Z}^n$ *FC(V)* denota o fecho convexo de *V*. Para $V \subset C_g \cap \mathbb{Z}^n$ consideremos a função $f : P(C_g \cap \mathbb{Z}^n) \to \mathbb{N}$ definida como a quantidade de elementos que estão em C^g, não estão em *FC(V),* e estão em $\mathbb{Z}^n$, isto é,

$$f(V) = |(C_g - FC(V)) \cap \mathbb{Z}^n| \qquad (1)$$

onde "–" denota a diferença entre conjuntos, $|\cdot|$ a cardinalidade de um conjunto e *P(.)* o conjunto de partes.

Neste trabalho estudamos o seguinte problema de otimização combinatória:

$$\begin{array}{l} \min\ f(V) \\ \textbf{sujeito a}: V \in P(C_g \cap \mathbb{Z}^n) \end{array} \qquad (2)$$

De fato, para todo $g \in \mathbb{Z}^n$ o conjunto $C_g \cap \mathbb{Z}^n$ é não vazio e finito. A restrição garante que o fecho convexo da solução ótima do problema é maximal no tocante a que entre a hiperesfera e o complemento do fecho convexo não há elementos em $\mathbb{Z}^n$. Por isto o chamaremos de problema de Poliedro Maximal Inscrito numa Hiperesfera (PMIH).

Agora, denotemos *B(g, n)* o bloco $[-\lfloor \|g\| \rfloor, \lfloor \|g\| \rfloor]^n$ onde $\lfloor\cdot\rfloor$ denota a parte inteira inferior. Como $C_g \cap \mathbb{Z}^n \subset B(g,n) \cap \mathbb{Z}^n$, a cardinalidade de $C_g \cap \mathbb{Z}^n$ é menor ou igual que $(2\lfloor \|g\| \rfloor + 1)^n$, assim, temos $|C_g \cap \mathbb{Z}^n| \leqslant (2\lfloor \|g\| \rfloor + 1)^n$. Por outro lado, seja $H = \partial B(g,n) \cap C_g$, onde $\partial B(g,n)$ é a fronteira de *B(g, n)*. Como $FC(H) \cap \mathbb{Z}^n \subset C_g \cap \mathbb{Z}^n$, temos que

$$|FC(H) \cap \mathbb{Z}^n| \leqslant |C_g \cap \mathbb{Z}^n| \leqslant (2\lfloor \|g\| \rfloor + 1)^n. \tag{3}$$

Assim, temos que $2^{|FC(H) \cap \mathbb{Z}^n|} \leqslant 2^{|C_g \cap \mathbb{Z}^n|} \leqslant 2^{(2\lfloor \|g\| \rfloor + 1)^n}$. A partir desta desigualdade, podemos afirmar que o problema do PMIH pode ser resolvido no pior caso usando força bruta em $O(2^{(2\lfloor \|g\| \rfloor + 1)^n})$ A motivação central deste trabalho é obter o fecho convexo dos pontos em $\mathbb{Z}^n$ que satisfazem a restrição em (2) por meio de um algoritmo de ordem de tempo de execução menor que $O(2^{(2\lfloor \|g\| \rfloor + 1)^n})$.

3 I ALGORITMO

As entradas do Algoritmo 1 são $n \in \mathbb{N}^*$, a dimensão onde está a hiperesfera, e $g \in \mathbb{Z}^n$ um ponto que define o raio $r = \|g\| > 0$ da hiperesfera. A saída do Algoritmo 1 é um conjunto dos pontos em $\mathbb{Z}^n$ mais próximos da hiperesfera que define C_g.

Cada um destes pontos será implementado por uma tupla, cuja dimensão crescem à medida que novas dimensões forem inseridas. Inicialmente, cada tupla tem dimensão 1 e será representada por (k), onde $k \in \mathbb{Z}$. A dimensão de uma tupla pode ser incrementada da seguinte forma. Seja *I* uma tupla de dimensão *n – 1* tal que $I = (i_1, i_2, ..., i_{n-1})$. Acrescenta-se uma coordenada em *I*, simbolizada no Algoritmo 1, por *(I, k)*, para um valor qualquer $k \in \mathbb{Z}$. Uma tupla, portanto, pode ser implementada com uma lista encadeada. No Algoritmo 1, a saída está implementada como uma deque de tuplas.

Dado $\boldsymbol{x} = (x_1, \ldots, x_{n-1}) \in \mathbb{Z}^{n-1}$, o *resíduo de norma inteiro* é o maior inteiro *s* tal que $\|(\boldsymbol{x}, s)\| = \|(x_1, \ldots, x_{n-1}, s)\| \leqslant r$. Em outras palavras, o resíduo de norma inteiro define o maior valor inteiro que se permite inserir na tupla $\boldsymbol{x}$ sem aumentar a norma de *(**x**, s)* além do valor *r*. Para a norma euclidiana, o resíduo de norma inteiro é dado por $s = \pm \lfloor \sqrt{r^2 - \|x\|^2} \rfloor$. A ideia de resíduo de norma inteiro pode ser definida usando qualquer norma e assim o Algoritmo 1 pode ser adaptado para bolas n-dimensionais definidas com outra norma, basta mudar a equação do resíduo de norma inteiro na linha 15 do Algoritmo 1.

3.1 Corretude

O Algoritmo 1 inicialmente verifica o caso trivial, isto é, quando n = 1, e executa apenas as linhas 3 e 4, pois a iteração da linha 10 a 29 não é executada neste caso. Para o laço externo nas linhas 10 a 29, tem-se o seguinte invariante de laço (CORMEN et al., 2009):

$$\left(2 \leqslant i < n \Rightarrow R = \{(\boldsymbol{x}, k) \in \mathbb{Z}^{i-1} : \|(\boldsymbol{x}, k)\| \leqslant r \wedge k \in \{-s, -s+1, \ldots, s-1, s\} \wedge \right.$$
$$\left. s \text{ é resíduo de norma inteiro de } \boldsymbol{x}\}\right) \wedge (i \geq n \Rightarrow R = \{\boldsymbol{x} \in \mathbb{Z}^n : r-1 < \|\boldsymbol{x}\| \leqslant r\}).$$

Este invariante se mantém nas três fases do laço. Na **inicialização**, após a atribuição $i \leftarrow 2$, devido a execução prévia das linhas 6 a 8, tem-se que

$$R = \{(\boldsymbol{x}, k) \in \mathbb{Z} : \|(k)\| \leqslant r \wedge k \in \{-s, -s+1, \ldots, s-1, s\}\}.$$

Na **manutenção**, o laço das linhas 12 a 28 adiciona uma coordenada a mais em cada elemento de *R*. Cada elemento de *R* com dimensão *i – 1* será removido (linha 13) para serem adicionados novos elementos em *R* com dimensão *i*. O resíduo de norma inteiro para este elemento removido é calculado na linha 15. Os valores a serem adicionados na tupla dependem do valor de i, testado na condição da linha 16. Se $i \neq n$, as linhas de 24 a 26 são executadas e, portanto, após a linha 28,

$$R = \{(\boldsymbol{x}, k) \in \mathbb{Z}^i : \|(\boldsymbol{x}, k)\| \leqslant r \wedge k \in \{-s, -s+1, \ldots, s-1, s\}\}.$$

Se $i = n$, as linhas 16 a 22 são executadas e tem-se

$$R = \{\boldsymbol{x} \in \mathbb{Z}^n : r-1 < \|\boldsymbol{x}\| \leqslant r\}.$$

Quando *i* é incrementado na linha 29, o invariante se mantém.

No **término**, tem-se que $i = n+1$. O invariante é verdadeiro devido ao segundo condicional. Assim, implica-se que $R = \{\boldsymbol{x} \in \mathbb{Z}^n : r-1 < \|\boldsymbol{x}\| \leqslant r\}$.

Algoritmo 1: Lista os pontos inteiros internos a uma esfera n-dimensional mais próximos da sua superfície.

Entrada: $n \in \mathbb{N}^*, g \in \mathbb{Z}^n$
Saída: $R = \{x \in \mathbb{Z}^n : r-1 < \|x\| \leqslant r\}$
1 Inicialize uma deque *R* vazia;
2 **se** n = 1 **então**
3   Adicione a *R* a tupla $(-\lfloor r \rfloor)$;
4   Adicione a *R* a tupla $(\lfloor r \rfloor)$;
5 **senão**
6   **para** $k \leftarrow -\lfloor r \rfloor$ **até** $\lfloor r \rfloor$ **faça**
7   Adicione a *R* a tupla (k);
8   **fim**
9 **fim**
10 **para** $i \leftarrow 2$ **até** n **faça**
11   Atribua a *Rl* o tamanho atual de *R*;
12   **para** $j \leftarrow 1$ **até** Rl **faça**
13   Atribua a *l* a tupla no topo de *R* e desempilhe-a de *R*;
14         $l \leftarrow \|l\|$;
15         $s \leftarrow \lfloor \sqrt{r^2 - l^2} \rfloor$;
16   **se** n = i **então**

```
17          se s ≠ 0 então
18              Adicione no final da deque R a tupla (I, –s);
19              Adicione no final da deque R a tupla (I, s);
20          senão
21              Adicione no final da deque R a tupla (I, s);
22          fim
23          senão
24          para k □ – s até s faça
25              Adicione no final da deque R a tupla (I, k);
26          fim
27          fim
28      fim
29  fim
```

3.2 Complexidade de Tempo E Espaço

Quando *n = 1*, o algoritmo executa em tempo constante as adições de apenas dois elementos na deque. Para *n > 1*, a complexidade do algoritmo quanto a adições de elementos em *R* é dada por:

$$\underbrace{2\lfloor r \rfloor + 1}_{\text{linhas 6 a 8}} + \underbrace{E(j)|R_{n-1}|}_{\text{linhas 16 a 22}} + \underbrace{\sum_{i=2}^{n-1} \sum_{j=1}^{|R_{i-1}|} \left(2\left\lfloor \sqrt{r^2 - \| I_j \|^2} \right\rfloor + 1\right)}_{\text{linhas 24 a 26}} \quad (4)$$

com R_i representando a deque de dimensão *i*, I_j denotando a tupla *j* existente na deque R_i e *E(j) = 1* se para $\| I_j \|$, $s = 0$ na linha 15 e *E(j) = 2,* caso contrário. Os valores de $\| I_j \|$ são limitados por *r* e o tamanho de cada R_i é limitado por $(2\lfloor r \rfloor + 1)^i$, devido à Equação 3. Assim, o algoritmo executa em tempo $O(r^{n-1})$.

Quanto a complexidade de espaço, note que a quantidade de adições a *R* é o mesmo que a quantidade de elementos armazenados nesse conjunto multiplicada pelo tamanho de cada tupla. Logo, incluindo as variáveis usadas nos laços (*k, i, j, Rl, l, s*) que permanece constante, a quantidade de espaço usada no algoritmo também é $O(nr^{n-1})$.

A solução do problema PMIH é obtida usando a saída do Algoritmo 1 como entrada do algoritmo proposto em (CHAZELLE, 1993) para retornar o fecho convexo destes pontos. Este algoritmo roda em tempo $O(r^{n-1} \log r^{n-1} + r^{(n-1)\lfloor n/2 \rfloor})$.

REFERÊNCIAS

CHAZELLE, B. An optimal convex hull algorithm in any fixed dimension. **Discrete & Computational Geometry**, v. 10, p. 377–409, 1993.

CORMEN, T. H. et al. **Introduction to Algorithms**. 3rd. ed. Massachusetts: The MIT Press, 2009.

LOZADA, E. G. M. et al. PMIH: Poliedro Maximal de Vértices em Zn Inscrito em uma Hiperesfera em Rn. In: **Anais do XXXVII congresso da sociedade brasileira de computação**. São Paulo – SP: Sociedade Brasileira de Computação, 2017.

MALTEZ, T. R. R. **Polıgono Inscrito em uma Circunferência sem pontos Z2 entre a Circunferência e o Polıgono**. 2016. Monografia (Bacharel em Ciências Exatas e Tecnológicas), UFRB (Universidade Federal do Recôncavo da Bahia), Cruz das Almas, Brasil.

MALTEZ, T. R. R.; LOZADA, E. G. M. Uma ferramenta dinâmica de geogebra para o estudo de polıgonos não regulares inscritos em uma circunferência. In: **III Reconcitec**. Cruz das Almas – BA: UFRB, 2015.

MOTZKIN, T. et al. The double description method. In: KUHN, H. W.; TUCKER, A. W. (Ed.). **Contributions to the Theory of Games II.** Princeton: Princeton University Press, 1953.

ROCKAFELLAR, R. **Convex Analysis**. Princeton: Princeton University Press, 1997. (Princeton landmarks in mathematics and physics).

SCHRIJVER, A. **Theory of Linear and Integer Programming**. New York, NY, USA: John Wiley and Sons, Inc., 1986. ISBN 0-471-90854-1.

SOBRE O ORGANIZADOR

Ernane Rosa Martins - Doutorado em andamento em Ciência da Informação com ênfase em Sistemas, Tecnologias e Gestão da Informação, na Universidade Fernando Pessoa, em Porto/Portugal. Mestre em Engenharia de Produção e Sistemas pela PUC-Goiás, possui Pós-Graduação em Tecnologia em Gestão da Informação pela Anhanguera, Graduação em Ciência da Computação pela Anhanguera e Graduação em Sistemas de Informação pela Uni Evangélica. Atualmente é Professor de Informática do Instituto Federal de Educação, Ciência e Tecnologia de Goiás - IFG (Câmpus Luziânia), ministrando disciplinas nas áreas de Engenharia de Software, Desenvolvimento de Sistemas, Linguagens de Programação, Banco de Dados e Gestão em Tecnologia da Informação. Pesquisador do Núcleo de Inovação, Tecnologia e Educação (NITE).

Agência Brasileira do ISBN
ISBN 978-85-7247-157-2
9 788572 471572