

DEMONSTRAÇÕES

0.1 Teorema de Convergencia

Teorema 1. *O algoritmo $AIOE(Q_n(a))$ encontra uma solução ótima para o problema $IOE(Q_n(a))$*

Prova. Primeiro, consideraremos o caso em que $n = 2$. Para $a \in \mathbb{R}_{++}^2$, seja $w = (w_1, w_2) \in E(Q_2(a))$ uma solução ótima do problema $IOE(Q_n(a))$. Primeiro, vamos supor, sem perda de generalidade, que $w_1 \geq w_2$, o que significa que $w_1 \geq \lfloor a_1 \rfloor - 1$. Portanto, para algum $k \in \mathbb{N}$, com $\lfloor a_1 \rfloor - 1 \leq k \leq \lfloor a_1 \rfloor + 1$, ele vale $g_1^k = w_1$. Como w é ótimo para o problema $IOE(Q_n(a))$ e vale $c \in \mathbb{R}_+^n$, segue-se que $w_2 \geq g_2^k$. Por outro lado, da definição da função g_a , segue-se que $w_2 \leq g_2^k$. Portanto, concluímos que $g^k = w$ e a afirmação é válida.

A seguir provaremos o caso geral., seja $w = (w_1, \dots, w_n)$ uma solução ótima do problema $IOE(Q_n(a))$. Claramente, a partir da definição da função $f_{\bar{a}}$ para $\bar{a} = a^k$ como no algoritmo, e o fato de que w em $E(Q_n(a)) \cap \mathbb{Z}_+^{n+}$, segue que $w_1 \leq \lfloor a_1 \rfloor$ e $w_k \leq f_{(a_1, \dots, a_k)}(w_1, \dots, w_{k-1})$ para $k = 1, \dots, n-2$. Estas relações implicam que, para $k = 1, \dots, n-2$, o índice i_k assumirá eventualmente o valor w_k . A seguir, analisaremos as iterações do ciclo mais interno do algoritmo quando $i_k = w_k$ para $k = 1, \dots, n-2$. Primeiro, observe que apenas os dois últimos componentes dos vetores w^k mudam ao longo dessas iterações. Além disso, cálculos simples mostram que nestas duas componentes temos iterações do algoritmo $AIOE(Q_n(a))$ no caso particular quando $n = 2$, aplicado à resolução do problema $\max\{(c_{n-1}, c_{n-2})x : x \in E(Q_n(\bar{a})) \cap \mathbb{Z}_+^2\}$, com

$$\bar{a} = \left(1 - \sum_{k=1}^{n-2} \left(\frac{i_k}{a_k}\right)^2\right)^{\frac{1}{2}} (a_{n-1}, a_n)$$

Agora, segue-se da primeira parte da prova que estas iterações resultarão no cálculo das soluções deste problema relacionado. Em particular, obteremos o par (w_{n-1}, w_n) , que é ótimo porque completa a solução ótima w . Em resumo, o algoritmo encontrará a solução ótima w no final do ciclo mais interno considerado acima. \square

0.2 Teorema da Complexidade

Lema 1. *Seja $a \in \mathbb{R}_{++}^n$ tal que $a_i > a_{i+1}$, para $i \in \{1, 2, \dots, n-1\}$ e consideremos $f_a : E(Q_n(a)) \rightarrow \mathbb{R}_+$ definida como*

$$f_a(x) = \left\lfloor a_n \left(1 - \sum_{i=1}^{n-1} \left(\frac{x_i}{a_i}\right)^2\right)^{\frac{1}{2}} \right\rfloor$$

então $f_a(x) < \lfloor a_1 \rfloor$ para todo $x \in E(Q_n(a))$.

Prova. Como $x \in E(Q_n(a)) \Rightarrow x^t Q_n(a) x \leq 1 \Leftrightarrow \sum_{i=1}^n \left(\frac{x_i}{a_i}\right)^2 \leq 1$ logo, $\sum_{i=1}^{n-1} \left(\frac{x_i}{a_i}\right)^2 < 1$, ou seja, $1 - \sum_{i=1}^{n-1} \left(\frac{x_i}{a_i}\right)^2 < 1$, sabemos que

$$f_a(x) = \left\lfloor a_n \left(1 - \sum_{i=1}^{n-1} \left(\frac{x_i}{a_i}\right)^2\right)^{\frac{1}{2}} \right\rfloor$$

com efeito, $f_a(x) < \lfloor a_n \rfloor$ e por hipótese $a_1 > a_2 > \dots > a_n$, portanto $f_a(x) < \lfloor a_1 \rfloor$ para todo $x \in E(Q_n(a))$ \square

Lema 2. *Seja $a \in \mathbb{R}_{++}^n$. O número de operações aritméticas requeridas para calcular $\left\lfloor a_{n-1} \left(1 - \sum_{l=1}^{n-2} \left(\frac{i_l}{a_l}\right)^2\right)^{\frac{1}{2}} \right\rfloor$ é $2n$,*

Prova.. 2 operações são realizadas para calcular cada termo $\left(\frac{i_l}{a_l}\right)^2$, logo, para o calculo da $\sum_{l=1}^{n-2} \left(\frac{i_l}{a_l}\right)^2$, serão realizadas $2(n-2)$ operações. Depois devemos realizar uma subtração, o calculo da uma raiz quadrada, para logo fazer um produto e finalizar com o calculo da parte inteira, o que resulta em 4 operações, totalizando $2n$ operações aritméticas. \square

Teorema 2. *Para encontrar uma solução para o problema $IOE(Q_n(a))$, o algoritmo $AIOE(Q_n(a))$ é da $O(na_1^{n-1})$.*

Prova. Assumiremos que $a_i > a_{i+1}$, para $i \in \{1, \dots, n-1\}$. Usando o lema 2 podemos afirmar que para calcular w^k na linha 19 o algoritmo executa $2n + 5$ operações. Para o calculo de p^* , na linha 20, são feitas $4n - 1$. Por outra parte, pelo lema 1 e pela linha 16, $s < \lfloor a_1 \rfloor + 1$, assim o ciclo definido entre a linha 18 e 21 são realizadas $(6n + 4)(\lfloor a_1 \rfloor + 1)$. Observe que o calculo de w^0 na linha 17, depende do calculo de s na linha 16, por tanto usando o lema 2, o algoritmo nestas duas linha realiza $2n + 1$ operações. Usando o lema 1 nos ciclos definidos nas linhas 11 a 15 podemos afirmar no pior caso o algoritmo realize

$$\begin{aligned} & \sum_{i_1=0}^{\lfloor a_1 \rfloor} \dots \sum_{i_{n-2}=0}^{\lfloor a_1 \rfloor} (6n + 4)(\lfloor a_1 \rfloor + 1) \\ &= (6n + 4)(\lfloor a_1 \rfloor + 1) \sum_{i_1=0}^{\lfloor a_1 \rfloor} \dots \sum_{i_{n-2}=0}^{\lfloor a_1 \rfloor} 1 \\ &= (6n + 4)(\lfloor a_1 \rfloor + 1)^{n-1} < (6n + 4)(a_1 + 1)^{n-1} \end{aligned}$$

Por tanto o algoritmo é da $O(na_1^{n-1})$ \square