Capítulo 7

Decomposição de Elementos Estruturantes Arbitrários

Este capítulo apresenta um algoritmo geral para decomposição de EE's arbitrários. Com este algoritmo é possível verificar se uma erosão (respectivamente, dilatação) arbitrária tem uma decomposição seqüencial. Se uma decomposição de um EE A existe, então é apresentada uma decomposição ótima de A. O algoritmo usa a técnica de "branch and bound", com estratégias de poda baseadas em propriedades algébricas e geométricas da adição de Minkowski. A técnica aqui apresentada generaliza resultados clássicos como os de Zhuang and Haralick [51], Xu [49], e Park and Chin [35], com um desempenho melhor ou equivalente. Finalmente, análises teóricas do algoritmo proposto e resultados experimentais também são apresentados.

7.1 Introdução

Neste capítulo, apresentaremos um algoritmo para encontrar, se existe, uma decomposição de um EE arbitrário. Se uma tal decomposição existe, o algoritmo fornece uma decomposição com um número mínimo de elementos; caso contrário, ele mostra que o EE não tem uma decomposição.

Como no trabalho de Zhuang and Haralick [51], a idéia fundamental do método proposto é a aplicação de técnicas de Otimização Combinatória. A formulação adotada no algoritmo proposto neste capítulo é uma busca baseada em "branch and bound" em uma árvore que representa o espaço de todas as possíveis seqüências de subconjuntos que poderiam ser uma decomposição do EE de entrada. Nesta busca, o algoritmo procura soluções válidas, podando as soluções não viavéis. A eficiência de encontrar tais podas, que é baseada em propriedades algébricas e geométricas da adição de Minkowski, é essencial para a viabilidade do método. Se nenhuma solução válida é encontrada, então o EE não tem decomposição.

7.1.1 O Método "Branch and Bound"

No Capítulo 4, Seção 4.1, vimos que o problema de encontrar uma decomposição de um EE A equivale ao problema de encontrar uma subseqüência de $\mathsf{SeqInv}[A]$.

Dado um EE A tal que $\mathsf{SeqInv}[A]$ tenha um número grande de elementos, não é possível enumerar, em um tempo razoável, todas subseqüências de $\mathsf{SeqInv}[A]$. Isto evidencia o fato de que não é possível encontrar uma decomposição de A de comprimento mínimo por uma enumeração explícita de todas soluções possíveis. Assim, o problema de decompor um EE arbitrário é um problema de Otimização Combinatória.

Um método para enumerar parcialmente as possíveis soluções de um problema de Otimização Combinatória é conhecido como *"branch and bound"* [37, p. 40]. A idéia é particionar o conjunto de soluções possíveis em alguns conjuntos pequenos, gerando subproblemas independentes. Quando se resolve todos estes subproblemas, a melhor solução encontrada é a solução ótima do problema original. Esta partição pode ser vista como a construção de uma árvore, cujos nós correspondem às soluções parciais, e o nó raiz ao problema original. A busca nesta árvore pode ser melhorada se limites inferior e superior do valor da solução ótima são conhecidos. Por exemplo, se é conhecido uma solução viável cujo valor é, digamos, 25 e o nó atualmente visitado tem limite inferior igual a 28, este ramo pode ser podado, uma vez que a melhor solução neste ramo não pode ser melhor que a melhor solução conhecida. A arte desta estratégia consiste em encontrar bons limites para evitar de visitar todas (ou a grande maioria) possíveis soluções do problema.

Há muitas maneiras de visitar os nós de uma árvore. No algoritmo apresentado na Seção 7.2, usamos a busca em profundidade [37, p. 39], que é descrita da seguinte forma. No início da busca, cada nó da árvore é marcado como "não visitado". Na primeira iteração, algum nó arbitrário é selecionado (se a árvore tem uma raiz, então a raiz é selecionada) e marcado como "visitado". Seus nós adjacentes são marcados como "alcançável" e colocados em uma pilha. Em uma iteração arbitrária, se a pilha está vazia, a busca termina. Caso contrário, o elemento que está no topo da pilha é desempilhado e marcado como "visitado". Os seus nós adjacentes que ainda estão marcados como "não visitado" são marcados como "alcançável" e colocados na pilha e uma nova iteração começa.

7.1.2 A Árvore de Decomposição

Dado um EE A, definimos uma árvore rotulada que representa o espaço de todas possíveis subseqüências de $\mathsf{SeqInv}[A]$. Seja $[B_1, B_2, \cdots, B_m]$ a seqüência de invariantes de A, isto é, $\mathsf{SeqInv}[A] = [B_1, B_2, \cdots, B_m]$. A *árvore de decomposição* de A, denotada $\mathcal{T}(A)$, é uma árvore rotulada tal que:

(1) Todos nós são rotulados por um subconjunto $Y = B_{i_1} \oplus B_{i_2} \oplus \cdots \oplus B_{i_j}$, onde $[B_{i_1}, B_{i_2}, \cdots, B_{i_j}]$ é uma subseqüência de $\mathsf{SeqInv}[A]$.

(2) O rótulo da raiz é o conjunto unitário que contém a origem e é denotado por $\{o\}$;

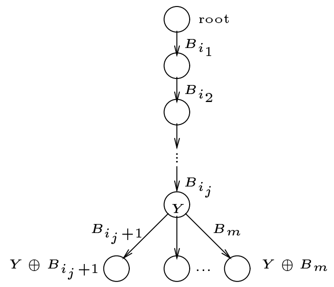

Figura 7.1: Um nó Y na árvore de decomposição.

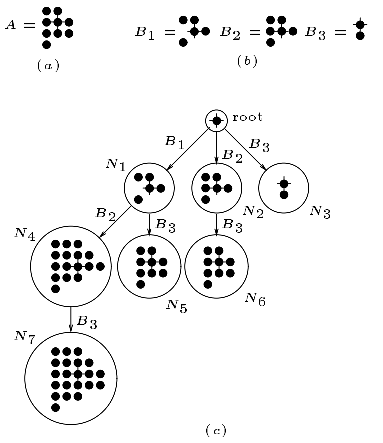

Figura 7.2: (a) Um EE A. (b) Seqüência de invariantes de A. (c) Árvore de decomposição de A.

(3) Os rótulos dos filhos de um nó cujo rótulo é $Y = B_{i_1} \oplus B_{i_2} \oplus \cdots \oplus B_{i_j}$ são $Y \oplus B_{i_j+1}$ (primeiro filho), $Y \oplus B_{i_j+2}$ (segundo filho), $\cdots, Y \oplus B_m$ (último filho). Para um exemplo, veja a Figura 7.1.

(4) A aresta que liga um nó cujo rótulo é Y e seu filho cujo rótulo é $Y \oplus B_k$ é rotulada como B_k. (veja a Figura 7.1).

Freqüentemente, usaremos nó Y significando nó cujo rótulo é Y.

Por construção da árvore de decomposição, não é difícil ver que, se a seqüência de invariantes de um dado EE A tem m elementos, então $\mathcal{T}(A)$ tem 2^m nós. Para um exemplo de $\mathcal{T}(A)$, veja a Figura 7.2.

Seja A um EE e sejam X e Y dois nós de $\mathcal{T}(A)$ tais que X é um descendente de Y. Denotamos $\mathsf{Path}_A[Y, X]$ a seqüência de rótulos das arestas de $\mathcal{T}(A)$ que estão no caminho de Y a X. Não é difícil ver que se $\mathsf{Path}_A[Y, X] = [C_1, C_2, \cdots, C_j]$, então $X = Y \oplus C_1 \oplus \cdots \oplus C_j$. Por exemplo, na Figura 7.2.c, $\mathsf{Path}_A[N_1, N_7] = [B_2, B_3]$ e $N_7 = N_1 \oplus B_2 \oplus B_3$.

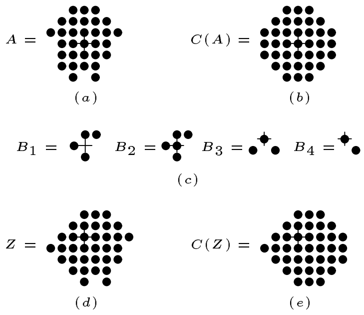

Figura 7.3: $(a) - (b)$ Um EE A e $C(A)$. (c) Seqüência de invariantes de A. $(d) - (e)$ Z e $C(Z)$.

Segundo esta notação, $\mathsf{Path}_A[\{o\}, Y]$ denota a seqüência de rótulos das arestas de $\mathcal{T}(A)$ que estão no caminho da raiz ao nó Y, que por sua vez forma uma subseqüência de $\mathsf{SeqInv}[A]$ cujo resultado da adição de Minkowski de todos elementos desta subseqüência é igual a Y. Por exemplo, na Figura 7.2.c, $\mathsf{Path}_A[\{o\}, N_6] = [B_2, B_3]$ é uma subseqüência de $\mathsf{SeqInv}[A]$; e o nó $N_6 = B_2 \oplus B_3$. Assim, se o EE A é igual a uma translação do nó Y (isto é, $Y \equiv A$), então A tem uma decomposição dada pela seqüência $\mathsf{Path}_A[\{o\}, Y]$.

O *nível* de um nó Y na árvore de decomposição é o comprimento do caminho da raiz até Y e é denotado por $level(Y)$. Desta forma, dado um EE A, o nosso objetivo é encontrar um nó Y em $\mathcal{T}(A)$ tal que $Y \equiv A$ e seu nível seja mínimo. Se tal nó não existe, então o EE A não é decomponível.

Seja A um EE. Dado um nó Y de $\mathcal{T}(A)$, dizemos que Y é um *nó viável* se, e somente se, existe um nó X, descendente de Y, tal que $X \equiv A$. Por exemplo, na Figura 7.2.c, o nó N_1 é um nó viável, enquanto que o nó N_4 não é.

A arte da estratégia do algoritmo apresentado na Seção 7.2 é visitar os nós da árvore de decomposição, procurando por nós viáveis e "podando" os nós inviáveis.

7.1.3 Condições Necessárias para Nós Viáveis

Nesta subseção, mostraremos algumas condições necessárias para a existência de nós viáveis na árvore de decomposição de um dado EE A.

O resultado a seguir fornece uma condição necessária para a existência de uma decomposição para um dado EE.

Teorema 7.1 *Seja A um EE. Seja Z o EE obtido pela adição de Minkowski de todos os elementos da seqüência* $\mathsf{SeqInv}[A]$. *Se existe $i \in \{0, \cdots, 7\}$ tal que $\nu_i(Z) < \nu_i(A)$, então A não tem decomposição.*

Prova: Suponha, por absurdo, que A tem uma decomposição. Pela Proposição 4.3, existe uma subseqüência de $\mathsf{SeqInv}[A]$, digamos $[B_1, B_2, \cdots, B_k]$, que é uma decomposição de A. Logo, existe $h \in \mathbb{Z}^2$ tal que $A = (B_1 \oplus B_2 \oplus \cdots \oplus B_k)_h$. Assim, pela Proposição 4.16, $\nu(A) = \nu(B_1) + \nu(B_2) + \cdots + \nu(B_k)$. Conseqüentemente, $\nu_j(A) = \nu_j(B_1) + \nu_j(B_2) + \cdots + \nu_j(B_k) \leq \nu_j(Z)$, para todo $j \in \{0, 1, \cdots, 7\}$, uma vez que Z é o resultado da adição de Minkowski de todos os elementos da seqüência $\mathsf{SeqInv}[A]$. Mas isto contradiz a hipótese de que existe $i \in \{0, 1, \cdots, 7\}$ tal que $\nu_i(A) > \nu_i(Z)$. Portanto, A não tem decomposição. ∎

A Figura 7.3.a apresenta um exemplo de um EE A que não tem decomposição. A seqüência de invariantes de A está apresentada na Figura 7.3.c. Seja Z o EE obtido pela adição de Minkowski de todos elementos em $\mathsf{SeqInv}[A]$, isto é, $Z = B_1 \oplus B_2 \oplus B_3 \oplus B_4$ (veja a Figura 7.3.d). O vetor de projeções de A e Z são, respectivamente, $\nu(A) = (2, 2, 2, 2, 2, 2, 2, 2)$ e $\nu(Z) = (2, 2, 2, 2, 2, 3, 0, 3)$. Assim, pelo Teorema 7.1, A não tem decomposição, uma vez que $\nu_6(Z) < \nu_6(A)$.

O próximo teorema, uma conseqüência da Proposição 4.16, fornece uma condição para nós viáveis, e, portanto, podemos utilizá-la como uma estratégia para podar alguns nós não viáveis na árvore de decomposição de um dado EE.

Teorema 7.2 *Seja A um EE. Seja Y um nó de $\mathcal{T}(A)$. Se existe $i \in \{0, \cdots, 7\}$ tal que $\nu_i(Y) > \nu_i(A)$, então Y não é um nó viável.*

Prova: Suponha por absurdo que Y é um nó viável. Assim, existe um descendente de Y, digamos X, tal que $X \equiv A$. Seja $W \subseteq \mathbb{Z}^2$ o conjunto obtido pela adição de Minkowski de todos elementos em $\mathsf{Path}_A[Y, X]$. Assim, $X = Y \oplus W \equiv A$, e, conseqüêntemente, existe $h \in \mathbb{Z}^2$ tal que $A = (Y \oplus W)_h$. Logo, pela Proposição 4.16, $\nu(A) = \nu(Y) + \nu(W)$. Desta forma, para todo $j \in \{0, 1, \cdots, 7\}$, $\nu_j(A) = \nu_j(Y) + \nu_j(W)$, e, portanto, $\nu_j(A) \geq \nu_j(Y)$. Mas isto contradiz a hipótese de que existe $i \in \{0, 1, \cdots, 7\}$ tal que $\nu_i(A) < \nu_i(Y)$. logo, Y não é um nó viável. ∎

Na Figura 7.2.c, os vetores projeção de N_4 e A são, respectivamente, $\nu(N_4) = (2, 2, 0, 2, 2, 0, 4, 0)$ e $\nu(A) = (1, 1, 1, 1, 1, 0, 3, 0)$. Assim, pelo Teorema 7.2, N_4 não é um nó viável, uma vez que $\nu_0(N_4) > \nu_0(A)$.

O teorema seguinte, que é uma conseqüência imediata da Proposição 3.15, é uma outra estratégia de poda (que também foi obtida por Zhuang e Haralick [51]).

Teorema 7.3 *Seja A um EE. Seja Y um nó de $\mathcal{T}(A)$. Se Y não é um invariante de A, então Y não é um nó viável.*

As definições que se seguem são necessárias para se obter uma outra estratégia de poda (dada pelo Teorema 7.5).

Seja A um EE. Seja Y um nó de $\mathcal{T}(A)$. Denotaremos por $\mathsf{Arestas}_A[Y]$ a seqüência formada pelos rótulos das arestas que ligam Y a seus filhos, isto é, se Y é o nó raiz, então

$\mathsf{Arestas}_A[Y] = \mathsf{SeqInv}[A]$; se $Y = B_{i_1} \oplus \cdots \oplus B_{i_j}$, então $\mathsf{Arestas}_A[Y] = [B_{i_j+1}, B_{i_j+2}, \cdots, B_m]$ (veja a Figura 7.1). Por exemplo, na Figura 7.2.c, $\mathsf{Arestas}_A[N_1] = [B_2, B_3]$. Se X é um descendente de Y, então, por construção de $\mathcal{T}(A)$, a seqüência $\mathsf{Path}_A[Y, X]$ é uma subseqüência de $\mathsf{Arestas}_A[Y]$. Por exemplo, na Figura 7.2.c, $\mathsf{Path}_A[N_1, N_5] = [B_3]$ é uma subseqüência de $\mathsf{Arestas}_A[N_1] = [B_2, B_3]$.

Seja A um EE. Seja Y um nó de $\mathcal{T}(A)$. Definimos a subseqüência $\mathsf{Possível}_A[Y]$ da seqüência $\mathsf{Arestas}_A[Y]$ da seguinte forma. Seja $D_1, D_2, \cdots, D_k$ todos elementos distintos de $\mathsf{Arestas}_A[Y]$ tais que, para todo $i \in \{1, 2, \cdots, k-1\}$, D_i aparece antes de D_{i+1} em $\mathsf{Arestas}_A[Y]$. Seja m_i o número de ocorrências de D_i em $\mathsf{Arestas}_A[Y]$. Claramente, $\mathsf{Arestas}_A[Y] = \mathsf{Seq}[D_1, m_1] \cdots \mathsf{Seq}[D_k, m_k]$. Seja n_i o maior inteiro não negativo tal que $(Y \oplus n_i D_i)$ é um invariante de A. Seja $d_i = \min\{n_i, m_i\}$. A *seqüência possível* é definida como $\mathsf{Possível}_A[Y] = \mathsf{Seq}[D_1, d_1] \cdots \mathsf{Seq}[D_k, d_k]$. Por exemplo, na Figura 7.2.c, $\mathsf{Possível}_A[N_1] = B_3$, uma vez que $\mathsf{Arestas}_A[N_1] = B_2, B_3$, $N_1 \oplus B_2 = N_4$ não é um invariante de A e $N_1 \oplus B_3 = N_5$ é um invariante de A.

No próximo resultado temos uma propriedade interessante para seqüências possíveis. Como conseqüência desta propriedade, obtemos uma nova estratégia de poda dada pelo Teorema 7.5.

Lema 7.4 *Seja A um EE. Sejam X, Y dois nós de $\mathcal{T}(A)$ tais que X é um descendente de Y. Se $X \equiv A$, então $\mathsf{Path}_A[Y, X]$ é uma subseqüência de $\mathsf{Possível}_A[Y]$.*

Prova: Sejam $S_1, S_2, \cdots, S_k$ todos elementos distintos da seqüência $\mathsf{Path}_A[Y, X]$, tais que, para todo $i \in \{1, 2, \cdots, k-1\}$, S_i aparece antes de S_{i+1} em $\mathsf{Path}_A[Y, X]$. Se s_i é o número de ocorrências de S_i na seqüência $\mathsf{Path}_A[Y, X]$, então $\mathsf{Path}_A[Y, X] = \mathsf{Seq}[S_1, s_1] \cdots \mathsf{Seq}[S_k, s_k]$. Assim, $X = Y \oplus s_1 S_1 \oplus s_2 S_2 \oplus \cdots \oplus s_k S_k$. Para provar que $\mathsf{Path}_A[Y, X]$ é uma subseqüência de $\mathsf{Possível}_A[Y]$, temos que mostrar que:

(a) cada $S_i \in \mathsf{Possível}_A[Y]$;

(b) se d_i é o número de ocorrências de S_i na seqüência $\mathsf{Possível}_A[Y]$, então $s_i \leq d_i$;

(c) para todo $i \in \{1, \cdots, k-1\}$, S_i aparece antes de S_{i+1} em $\mathsf{Possível}_A[Y]$.

Uma vez que $X = Y \oplus s_1 S_1 \oplus \cdots \oplus s_k S_k \equiv A$, existe $z \in \mathbb{Z}^2$ tal que $A = Y \oplus s_1 S_1 \oplus \cdots \oplus s_k S_k \oplus \{z\}$. Desta forma, pela Proposição 3.15, $(Y \oplus s_i S_i)$ é um invariante de A e, pela definição de $\mathsf{Possível}_A[Y]$, $S_i \in \mathsf{Possível}_A[Y]$. Isto prova ($a$).

Seja n_i é o maior inteiro positivo tal que $(Y \oplus n_i S_i)$ é invariante de A. Seja m_i o número de ocorrências de S_i em $\mathsf{Arestas}_A[Y]$. Pela construção da seqüência $\mathsf{Possível}_A[Y]$, o número de ocorrências de S_i em $\mathsf{Possível}_A[Y]$ é $d_i = \min\{n_i, m_i\}$. Se $d_i = n_i$, então $s_i \leq n_i$, uma vez que $Y \oplus s_i S_i$ é invariante de A. Se $d_i = m_i$, then $s_i \leq m_i$, uma vez que $\mathsf{Path}_A[Y, X]$ é uma subseqüência de $\mathsf{Arestas}_A[Y]$. Isto prova (b).

Suponha por absurdo que existe $j \in \{1, 2, \cdots, k-1\}$ tal que S_{j+1} aparece antes de S_j em $\mathsf{Possível}_A[Y]$. Uma vez que a seqüência $\mathsf{Possível}_A[Y]$ é uma subseqüência de $\mathsf{Arestas}_A[Y]$, então S_{j+1} aparece antes de S_j em $\mathsf{Arestas}_A[Y]$. Uma vez que $\mathsf{Path}_A[Y, X]$ é uma subseqüência de $\mathsf{Arestas}_A[Y]$, então S_{j+1} aparece antes de S_j em $\mathsf{Path}_A[Y, X]$. Mas isto é uma contradição, uma vez que, por construção de $\mathsf{Path}_A[Y, X]$, S_j aparece antes de S_{j+1} em $\mathsf{Path}_A[Y, X]$. Portanto, para todo $i \in \{1, 2, \cdots, k-1\}$, S_i aparece antes de S_{i+1} na seqüência $\mathsf{Possível}_A[Y]$. Isto prova (c). ■

O teorema seguinte fornece uma nova estratégia de poda.

Teorema 7.5 *Seja A um EE. Seja Y um nó de $\mathcal{T}(A)$. Seja $Z \subseteq \mathbb{Z}^2$ o conjunto obtido pela adição de Minkowski de todos os elementos de $\mathsf{Possível}_A[Y]$. Se existe $i \in \{0, \cdots, 7\}$ tal que $\nu_i(Z) < \nu_i(A) - \nu_i(Y)$, então Y não é um nó viável.*

Prova: Suponha por absurdo que Y é um nó viável. Assim, existe um descendente de Y, digamos X, tal que $X \equiv A$. Seja $W \subseteq \mathbb{Z}^2$ o conjunto obtido pela adição de Minkowski de todos elementos em $\mathsf{Path}_A[Y, X]$. Dessa forma, $X = Y \oplus W \equiv A$, e, conseqüentemente, existe $h \in \mathbb{Z}^2$ tal que $A = (Y \oplus W)_h$. Logo, pela Proposição 4.16, temos que $\nu(A) = \nu(Y) + \nu(W)$. Uma vez que, pelo Lema 7.4, a seqüência $\mathsf{Path}_A[Y, X]$ é uma subseqüência de $\mathsf{Possível}_A[Y]$, então, para todo $j \in \{0, \cdots, 7\}$, $\nu_j(Z) \geq \nu_j(W) = \nu_j(A) - \nu_j(Y)$. Mas, isto é uma contradição, pois, por hipótese, existe $i \in \{0, \cdots, 7\}$ tal que $\nu_i(Z) < \nu_i(A) - \nu_i(Y)$. Portanto, Y não é um nó viável. ■

Com as estratégias dadas pelos Teoremas 7.2, 7.3 e 7.5, podemos podar alguns (possivelmente, não todos) nós inviáveis da árvore de decomposição. Apresentamos na próxima seção o algoritmo que faz uso dessas propriedades de poda.

7.2 Decomposição Ótima de um Elemento Estruturante

Nesta seção, apresentaremos um algoritmo para encontrar uma decomposição ótima de um EE A arbitrário.

A proposição seguinte caracteriza um nó Y na árvore de decomposição de um dado EE A tal que $Y \equiv A$. Um resultado análogo, usando vetor retangular em vez do vetor projeção, foi apresentado na Proposição 5.7.

Proposição 7.6 *Seja A um EE e seja Y um nó de $\mathcal{T}(A)$. Então, $Y \equiv A$ se, e somente se, Y é um invariante de A e $\nu(Y) = \nu(A)$.*

Prova: $(\Rightarrow)$ Se $Y \equiv A$, então, existe $h \in \mathbb{Z}^2$ tal que $Y = A_h = A \oplus \{h\}$. Logo, pela Proposiçào 3.15, Y é um invariante de A. Uma vez que $\nu(A_h) = \nu(A)$, então, $\nu(Y) = \nu(A)$.

($\Leftarrow$) Uma vez que Y é um invariante de A, pela Proposição 3.14, existe um EE X tal que $A = X \oplus Y$. Pela Proposição 4.16, $\nu(A) = \nu(X) + \nu(Y)$. Uma vez que $\nu(A) = \nu(Y)$, então $\nu(X) = 0$ e, portanto, $|X| = 1$. Seja $h \in \mathbb{Z}^2$ tal que $X = \{h\}$. Neste caso, a adição de Minkowski $X \oplus Y = A$ é uma translação de Y por h. Logo, $Y \equiv A$. ■

Seja A um EE. Quando encontramos um nó Y de $\mathcal{T}(A)$ tal que Y é invariante de A e $\nu(Y) = \nu(A)$, então, pela Proposição 7.6, $Y \equiv A$, e, portanto, a seqüência $\mathsf{Path}_A[\{o\}, Y]$ é uma decomposição de A. Se $level(Y)$ é igual a $lower(A)$ (o limite inferior estabelecido pelo Teorema 4.18), então $\mathsf{Path}_A[\{o\}, Y]$ é uma solução ótima. Caso contrário, a solução ótima contém no máximo $level(Y)$ elementos. Assim, acabamos de obter um limite superior para a decomposição de A. Note que este limite superior pode mudar dinamicamente. O limite superior atual é denotado por lsa.

Uma tarefa crucial para o algoritmo é uma poda adequada na árvore de decomposição quando seus nós estão sendo visitados. O Teorema 7.1 fornece uma condição necessária para a existência de uma decomposição de A. Assim, devemos começar a busca verificando se esta condição é satisfeita. O Teorema 4.18 fixa o limite inferior para uma decomposição de A, enquanto os Teoremas 7.2, 7.3 e 7.5 guarantem podas importantes, uma vez que eles detectam nós inviáveis.

Neste contexto, o algoritmo verifica se a condição necessária para a existência de uma decomposição é satisfeita. Se ela não é satisfeita, então o EE não tem decomposição e o algoritmo pára. Caso contrário, depois de inicializar $lsa \leftarrow \infty$, a busca de uma decomposição começa. Na primeira iteração, o nó raiz é selecionado e marcado como "visitado" e seus filhos são colocados em uma pilha. Em uma iteração arbitrária, se a pilha está vazia, então o algoritmo verifica dois casos: (i) se $lsa = \infty$, então o EE A não tem decomposição; (ii) se $lsa = level(Y)$, então $\mathsf{Path}_A[\{o\}, Y]$ é uma decomposição ótima de A. Nos dois casos, a busca termina. Se a pilha não está vazia, o elemento do topo, digamos Y, é desempilhado e marcado como "visitado" e o algoritmo verifica se o nó Y pode ser podado ou se ele é uma solução, ou seja, se $Y \equiv A$.

(i) – *Verificar se o nó Y pode ser podado.*

Temos quatro estratégias de poda que devem ser examinadas:

$(i.a)$ *Poda por limite superior.*

Se $level(Y) \geq lsa$.

$(i.b)$ *Poda por projeção.*

Se existe i tal que $\nu_i(Y) > \nu_i(A)$.

$(i.c)$ *Poda por invariança.*

Se Y não é invariante de A.

$(i.d)$ *poda por seqüência possível.*

Seja $Z \subseteq \mathbb{Z}^2$ o conjunto obtido pela adição de Minkowski de todos os elementos da seqüência $\mathsf{Possível}_A[Y]$.

Se existe $i \in \{0, 1, \cdots, 7\}$ tal que $\nu_i(Z) < \nu_i(A) - \nu_i(Y)$.

Se uma dessas condições é satisfeita, então o nó Y é podado e começa-se uma nova iteração.

(ii) – *Verificar se o nó Y é uma solução.*

Y é uma solução se as duas condições seguintes são satisfeitas simultaneamente:

- $level(Y) \geq lower(A)$ e
- $\nu(Y) = \nu(A)$.

No caso dessas duas condições serem satisfeitas, Y é um invariante de A (caso contrário, Y seria podado no passo $(i.c)$) e, uma vez que $\nu(Y) = \nu(A)$, pela Proposição 7.6, $Y \equiv A$, e, portanto, $\mathsf{Path}_A[\{o\}, Y]$ é uma decomposição de A.

$(ii.a)$ – *O nó Y é uma solução.*

Temos duas possibilidades:

- se $level(Y) = lower(A)$
  $\Rightarrow$ A busca termina.
- se $level(Y) > lower(A)$
  $\Rightarrow lsa \leftarrow level(Y)$
  e começa-se uma nova iteração.

$(ii.b)$ – *O nó Y não é uma solução.*

Os filhos de Y são empilhados começa-se uma nova iteração.

Na Figura 7.4, apresentamos um exemplo simples de aplicação do algoritmo para encontrar uma decomposição ótima do EE A mostrado na Figura 7.4.a. A seqüência de invariantes de A é apresentada na Figura 7.4.b. O vetor projeção, o vetor retangular e o limite inferior de A estão mostrados na Figura 7.4.c. O nó raiz é selecionado para ser visitado na Figura 7.4.d. O nó N_1 está sendo visitado na Figura 7.4.e. O nó N_4 está sendo visitado na Figura 7.4.f. O nó N_4 é podado (poda por projeção, veja Teorema 7.2) na Figura 7.4.g. O nó N_5 está sendo visitado na Figura 7.4.h. Uma vez que $N_5 \equiv A$, então $\mathsf{Path}_A[\{o\}, N_5] = [B_1, B_3]$ é uma decomposição de A. Além disso, uma vez que $level(N_5) = 2 = lower(A)$, então $\mathsf{Path}_A[\{o\}, N_5]$ é uma solução ótima e a busca pára. Neste exemplo particular, somente a poda de projeção foi detectada. Em geral, as outras estratégias de podas são detectadas de uma maneira similar.

Note que o limite inferior, dado pelo Teorema 4.18, pode não ser atingido, ou seja, pode existir um EE A tal que o número de elementos em uma decomposição ótima de A é estritamente maior que $lower(A)$. Mas, se a entrada do algoritmo é um EE convexo, o Teorema 5.14, mostrado no Capítulo 5, guarante que este limite inferior sempre é alcançado.

A seguir faremos um comentário da complexidade do algoritmo proposto.

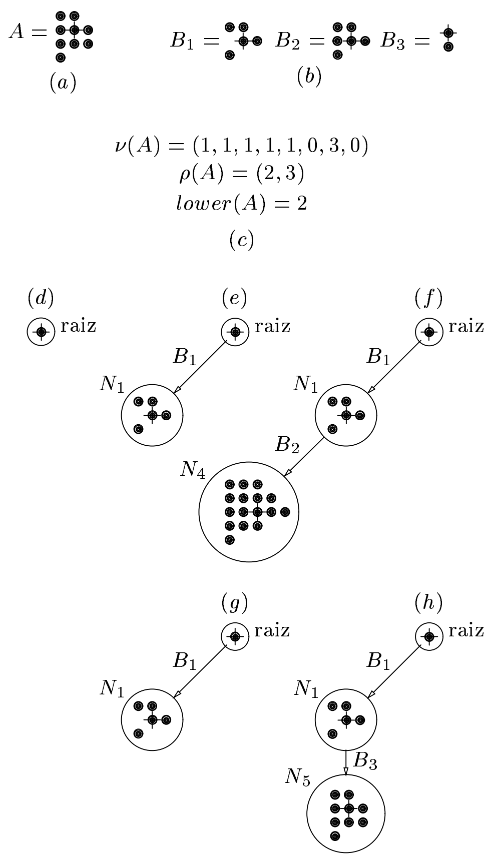

Figura 7.4: Aplicação do algoritmo para um exemplo simples.

Proposição 7.7 *Seja A um EE. Se m é o número de elementos na seqüência de invariantes de A, então $m = O(n)$, onde $n = \max\{\rho_0(A), \rho_1(A)\}$.*

Prova: Claramente, a multiplicidade de um dado EE com respeito a A é no máximo $\max\{\rho_0(A), \rho_1(A)\} = O(n)$ e o número de subconjuntos do quadrado elementar é 2^9. Assim, o número de elementos na seqüência $\mathsf{SeqInv}[A]$ é no máximo $2^9 \cdot \max\{\rho_1(A), \rho_2(A)\} = O(n)$. Portanto, $m = O(n)$. ■

Uma vez que a árvore de decomposição de um dado EE A contém 2^m nós, onde m é o número de elementos na seqüência de invariantes de A, e pela Proposição 7.7, $m = O(n)$, onde $n = \max\{\rho_0(A), \rho_1(A)\}$. Dessa forma, a complexidade do algoritmo proposto é, no pior caso, $O(2^n)$.

7.3 Comparação com Alguns Algoritmos Conhecidos

Nesta seção, compararemos o algoritmo apresentado na Seção 7.2 com alguns algoritmos conhecidos na literatura. A Subseção 7.3.1 compara com o algoritmo de Zhuang e Haralick [51] que decompõe EE's arbitrários. A Subseção 7.3.2 compara com o algoritmo do Xu [49] que decompõe EE's convexos. E a Subseção 7.3.3 compara com o algoritmo de Park e Chin [35] que decompõe EE's simplesmente conexos.

7.3.1 Algoritmo de Zhuang e Haralick

Zhuang e Haralick [51] apresentaram um algoritmo para encontrar uma decomposição ótima de um EE arbitrário onde todos elementos nesta decomposição têm um número k pré-fixado de pontos. O algoritmo deles faz uma busca em largura em uma árvore (que eles chamaram *árvore de busca* e a técnica de solução utilizada por eles é dividida em duas partes: (a) os pontos de um EE que participa de uma decomposição do EE de entrada A deve ser a diferença entre os pontos de A, isto é, se B está em uma decomposição de A, então, para todo $z \in B$, existem $x, y \in A$ tais que $z = x - y$; e (b) a redução do espaço de busca é feita usando a mesma estratégia dada pelo Teorema 7.3. Eles chamaram esta estratégia de *verificação para frente*.

A busca em largura tem vantagens e desvantagens em relação à busca em profundidade. A principal vantagem é que a primeira solução encontrada é sempre ótima. A principal desvantagem é que todos nós de um nível corrente devem estar guardados na memória. No algoritmo proposto, decidimos usar a busca em profundidade pois ele utiliza menos memória e, em nossos experimentos (veja a Seção 7.4), observamos que normalmente a distância entre a solução ótima e a primeira solução encontrada é pequena.

O problema de decomposição considerado neste trabalho (definido no Capítulo 3, Seção 3.8.1) é um caso especial do problema estudado por Zhuang e Haralick (isto é, $k = 2, 3, \cdots, 9$ e cada EE na decomposição é um subconjunto do quadrado elementar). Neste caso particular, o algoritmo proposto na Seção 7.2 consegue uma maior redução no espaço de busca, uma vez que ele usa mais duas estratégias de poda que o algoritmo de Zhuang e Haralick (podas por projeção e por seqüência possível).

7.3.2 Algoritmo do Xu

Xu [49] desenvolveu um algoritmo para encontrar uma decomposição ótima de EE's convexos em termos de subconjuntos do quadrado elementar, onde todos elementos nesta decomposição são também convexos (veja o Capítulo 5).

Dado um EE convexo, a complexidade do algoritmo do Xu é $O(n^2)$, onde n é o valor máximo entre as coordenadas do vetor retangular de A, ou seja, $= \max\{\rho_0(A), \rho_1(A)\}$. Nesta subseção, vamos mostrar que, se a entrada do nosso algoritmo é um EE convexo, então ele encontra uma decomposição ótima de A em tempo $O(n^4)$. Além disso, esta decomposição pode conter EE's não convexos (neste sentido, o algoritmo proposto é mais geral que o algoritmo do Xu).

Dado um EE A, dependendo da ordem escolhida para construir a seqüência SeqQ (ou equivalentemente, da ordem escolhida para os elementos de $\mathsf{SeqInv}[A]$), diferentes heurísticas de busca para se encontrar uma decomposição de A podem ser obtidas. Para este algoritmo, utilizamos a mesma ordem estabelecida no Capítulo 5, Seção 5.3, ou seja, em ordem decrescente segundo a soma das coordenadas dos vetores retangular de cada elemento em SeqQ (conseqüentemente em $\mathsf{SeqInv}[A]$), isto é, $\rho_0(B_1) + \rho_1(B_1) \geq \rho_0(B_2) + \rho_1(B_2) \geq \cdots \geq \rho_0(B_n) + \rho_1(B_n)$, e, ao mesmo tempo, se $\rho_0(B_i) + \rho_1(B_i) = \rho_0(B_j) + \rho_1(B_j)$ e $|B_i| \leq |B_j|$, então B_i aparece antes que B_j em SeqQ (conseqüentemente em $\mathsf{SeqInv}[A]$) (veja um exemplo na Figura 5.1).

Dado um EE A e um nó Y de $\mathcal{T}(A)$, seja D o primeiro elemento em $\mathsf{Arestas}_A[Y]$ tal que $Y \oplus D$ é um invariante de A. Definimos o nó $Y \oplus B$ como o *nó invariante mais à esquerda* de Y. Definimos a *seqüência de nós mais à esquerda* da árvore de decomposição $\mathcal{T}(A)$ como a seqüência $[T_0, T_1, T_2, \cdots, T_j]$ formada pelos nós de $\mathcal{T}(A)$ tais que T_0 é o nó raiz (isto é, um conjunto unitário que contém a origem) e, para todo $i = 1, 2, \ldots, j$, T_i é o nó invariante mais à esquerda de T_{i-1}. Dizemos que a seqüência de nós mais à esquerda $[T_0, T_1, T_2, \cdots, T_j]$ de $\mathcal{T}(A)$ é *maximal* se, e somente se, para todo elemento B de $\mathsf{Arestas}_A[T_j]$, $T_j \oplus B$ não é um invariante de A.

Agora, vamos mostrar que, dado um EE convexo A, se a seqüência de nós mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, T_2, \cdots, T_j]$, é maximal, então a seqüência $\mathsf{Path}_A[\{o\}, T_j] = [D_1, D_2, \cdots, D_j]$ é uma decomposição ótima de A. Para isso, dividiremos a nossa prova em duas partes. Primeiramente, vamos provar que $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição de A, ou seja, que $T_j \equiv A$ e, posteriormente, que $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição ótima, ou seja, $lower(A) = j$.

7.3.2.1 A Seqüência Maximal de Nós mais à Esquerda é uma Decomposição

Agora, vamos mostrar, no Teorema 7.12, que, dado um EE A, se a seqüência de nós mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, \cdots, T_j]$, é maximal, então a seqüência $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição de A. Para isso, precisamos de alguns resultados preliminares.

Lema 7.8 *Seja A um EE. Seja $[T_0, T_1, \cdots, T_j]$ a seqüência de nós mais à esquerda de $\mathcal{T}(A)$. Se $j > 0$ e $T_j \oplus D$ é um invariante de A, então $T_{j-1} \oplus D$ também é um invariante de A.*

Prova: Uma vez que $T_j \oplus D$ é um invariante de A, então, pela Proposição 3.14, existe $Z \in \mathbb{Z}^2$ tal que $(T_j \oplus D) \oplus Z = A$. Como $T_j = T_{j-1} \oplus D_j$, então $((T_{j-1} \oplus D_j) \oplus D) \oplus Z = A$. Pela propriedade de comutatividade e associatividade da adição de Minkowski temos que

$(T_{j-1} \oplus D) \oplus (D_j \oplus Z) = A$. E finalmente, pela Proposição 3.15, $T_{j-1} \oplus D$ é um invariante de A. ■

Lema 7.9 *Seja A um EE. Seja $[T_0, T_1, \cdots, T_j]$ a seqüência de nós mais à esquerda de $\mathcal{T}(A)$. Seja D um elemento da seqüência* $\mathsf{SeqInv}[A] = [B_1, B_2, \cdots, B_n]$. *Se $T_j \oplus D$ é um invariante de A, então D é um elemento de* $\mathsf{Arestas}_A[T_j]$.

Prova: Suponha por absurdo que $D \notin \mathsf{Arestas}_A[T_j]$. Seja m a multiplicidade de D com respeito a A. Como $\mathsf{Arestas}_A[T_0] = \mathsf{SeqInv}[A]$ e $D \in \mathsf{SeqInv}[A]$, então existe $i \in \{1, 2, \cdots, j\}$ tal que $\mathsf{Arestas}_A[T_{i-1}] = [B_a, B_{a+1}, \cdots, B_b] \cdot \mathsf{Seq}[D, m] \cdot [B_c, B_{c+1}, \cdots, B_n]$ e D é o primeiro elemento de $\mathsf{Arestas}_A[T_{i-1}]$ tal que $T_{i-1} \oplus D$ é um invariante de A, isto é, para todo elemento $B \in [B_a, B_{a+1}, \cdots, B_b]$, $T_{i-1} \oplus B$ não é invariante de A.

Seja $[D_1, D_2, \cdots, D_j]$ a seqüência dos rótulos das arestas que estão no caminho do nó raiz até o nó T_j. Assim, para cada $\ell \in \{1, 2, \cdots, j\}$, D_ℓ é o primeiro elemento de $\mathsf{Arestas}_A[T_{\ell-1}]$ tal que $T_{\ell-1} \oplus D_\ell$ é um invariante de A.

Uma vez que $T_j \oplus D$ é um invariante de A, então, podemos inferir pelo Lema 7.8 que, para todo $\ell \in \{1, 2, \cdots, j\}$, $T_{\ell-1} \oplus D$ é um invariante de A. Dessa forma, pela definição da seqüência de nós mais à esquerda, $D_i = D$, $D_{i+1} = D$, $\cdots$, $D_{i+m-1} = D$. Logo, $i + m - 1 \leq j$ e $T_i = T_{i-1} \oplus D_i = T_{i-1} \oplus D$, $T_{i+1} = T_i \oplus D_{i+1} = T_i \oplus D = T_{i-1} \oplus 2D$, $\cdots$, $T_{i+m-1} = T_{i-1} \oplus mD$, $T_{i+m} = T_{i-1} \oplus mD \oplus D_{i+m}$, $\cdots$, $T_j = T_{i-1} \oplus mD \oplus D_{i+m} \oplus \cdots \oplus D_j$. Portanto, se $Z = T_{i-1} \oplus D_{i+m} \oplus \cdots \oplus D_j$, então $T_j = Z \oplus mD$.

Como $T_j \oplus D$ é invariante de A, então, pela Proposição 3.14, existe um conjunto $T \in \mathbb{Z}^2$ tal que $(T_j \oplus D) \oplus T = A$. Como $T_j = Z \oplus mD$, então, $(Z \oplus T) \oplus (m+1)D = A$. Assim, pela Proposição 3.15, $(m+1)D$ é invariante de A. Mas isto contradiz a definição de multiplicidade de A. Logo, $D \in \mathsf{Arestas}_A[T_j]$ ■

Lema 7.10 *Seja A um EE convexo. A seqüência de nós mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, \cdots, T_j]$, é maximal se, e somente se, $\rho(T_j) = \rho(A)$.*

Prova: ($\Leftarrow$) Suponha por absurdo que a seqüência de nós mais à esquerda de $\mathcal{T}(A)$ não seja maximal. Seja B um elemento de $\mathsf{Arestas}_A[T_j]$ tal que $T_j \oplus B$ é um invariante de A.

Uma vez que $\mathsf{Arestas}_A[T_j]$ é uma subseqüência de $\mathsf{SeqInv}[A]$, então B tem pelo menos dois pontos e, conseqüentemente, $\rho_0(B) > 0$ ou $\rho_1(B) > 0$.

Uma vez que $T_j \oplus B$ é um invariante de A, então, pela Proposição 3.14, existe um EE X tal que $A = X \oplus (T_j \oplus B)$. Assim, pela Proposição 4.17, $\rho(A) = \rho(X) + \rho(T_j) + \rho(B)$. Como $\rho(T_j) = \rho(A)$, logo, $\rho(X) + \rho(B) = (0, 0)$ e, portanto, $\rho_0(B) = 0$ e $\rho_1(B) = 0$. Mas isto contradiz o fato de que $\rho_0(B) > 0$ ou $\rho_1(B) > 0$. Portanto, a seqüência de nós mais à esquerda de $\mathcal{T}(A)$ é maximal.

($\Rightarrow$) Suponha por absurdo que $\rho(T_j) \neq \rho(A)$. Obviamente, por construção da seqüência de nós mais à esquerda, T_j é um invariante de A.

Como A é um EE convexo e T_j é um invariante de A, então, pelo Lema 5.6, $A \ominus T_j$ é também um EE convexo. Assim, pela Proposição 5.1, $A \ominus T_j$ tem uma decomposição seqüencial, digamos $[Z_1, Z_2, \cdots, Z_\ell]$. Como T_j é um invariante de A, então, pela Proposição 5.19, $\rho(A \ominus T_j) = \rho(A) - \rho(T_j)$ Uma vez que $\rho(T_j) \neq \rho(A)$, então $\rho(A \ominus T_j) \neq (0,0)$. Assim, $|A \ominus T_j| \geq 2$, e, conseqüentemente, um elemento em $[Z_1, Z_2, \cdots, Z_\ell]$, digamos Z_1, contém pelo menos dois pontos. Claramente, Z_1 é um invariante de $A \ominus T_j$. Uma vez que Z_1 é um subconjunto do quadrado elementar e $|Z_1| \geq 2$, então existe um elemento B em $\mathsf{SeqInv}[A] = [B_1, B_2, \cdots, B_n]$ tal que $B \equiv Z_1$. Como Z_1 é um invariante de $A \ominus T_j$, obviamente, B também é um invariante de $A \ominus T_j$. Portanto, pela Proposição 3.14, existe um conjunto X tal que $B \oplus X = A \ominus T_j$.

Uma vez que, T_j é um invariante de A, então, $(A \ominus T_j) \oplus T_j = A$. Logo, $(B \oplus X) \oplus T_j = X \oplus (T_j \oplus B) = A$. Dessa forma, pela Proposição 3.15, $T_j \oplus B$ é um invariante de A. Portanto, pelo Lema 7.9, B é um elemento de $\mathsf{Arestas}_A[T_j]$. Mas isto contradiz a hipótese de que a seqüência de nós mais à esquerda de $\mathcal{T}(A)$ é maximal. Portanto, $\rho(T_j) = \rho(A)$.

■

Lema 7.11 *Seja A um EE convexo. A seqüência de nós mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, \cdots, T_j]$, é maximal, então $T_j \equiv A$.*

Prova: Por construção da seqüência de nós mais à esquerda, T_j é um invariante de A. Pelo Lema 7.10, $\rho(T_j) = \rho(A)$. Logo, pela Proposição 5.7, $T_j \equiv A$. ■

Teorema 7.12 *Seja A um EE convexo. A seqüência de nós mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, \cdots, T_j]$, é maximal, então a seqüência $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição de A.*

Prova: Seja $\mathsf{Path}_A[\{o\}, T_j] = [D_1, D_2, \cdots, D_j]$. Por construção, $T_j = D_1 \oplus D_2 \oplus \cdots \oplus D_j$. Pelo Lema 7.11, $T_j \equiv A$. Portanto, $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição de A. ■

7.3.2.2 A Seqüência Maximal de Nós mais à Esquerda é uma Decomposição Ótima

Agora, vamos provar que, dado um EE convexo A, se $[T_0, T_1, \cdots, T_j]$ é uma seqüência maximal de nós mais à esquerda de $\mathcal{T}(A)$, então $[T_0, T_1, \cdots, T_j]$ é uma decomposição ótima de A. Para isso, mostraremos que $lower(A) = j$. Como feito na Subseção 5.5.3, a Proposição 7.13 (cuja prova está na Seção 7.6) mostra que $[T_0, T_1, \cdots, T_j]$ é uma seqüência elemento retangular de A, e, dessa forma, pela Proposição 5.16, $lower(T_j) = j$. Como, pela Proposição 7.11, $T_j \equiv A$, então, claramente, $lower(A) = j$.

Proposição 7.13 *Se A é um EE convexo, então uma seqüência de nós mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, \cdots, T_j]$, é uma seqüência elemento retangular de A.*

O próximo resultado é uma conseqüência imediata das Proposições 7.13 e 5.16.

Proposição 7.14 *Seja A um EE convexo. Se a seqüência mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, T_2, \cdots, T_j]$, é maximal, então $lower(A) = j$.*

Prova: Pela Proposição 7.13, $[T_0, T_1, \cdots, T_j]$, é uma seqüência elemento retangular de A. Dessa forma, pela Proposição 5.16, $lower(T_j) = j$. Como, pelo Lema 7.11, $T_j \equiv A$, temos que, $\rho(T_j) = \rho(A)$, e, conseqüentemente, $lower(T_j) = lower(A)$. Portanto, $j = lower(T_j) = lower(A)$. ∎

Teorema 7.15 *Seja A um EE convexo. Se a seqüência mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, T_2, \cdots, T_j]$, é maximal, então $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição ótima de A.*

Prova: Pelo Teorema 7.12, $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição de A. Uma vez que $\mathsf{Path}_A[\{o\}, T_j]$ contém exatamente j elementos e, pela Proposição 7.14, $j = lower(A)$, então $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição ótima de A. ∎

7.3.2.3 Complexidade para Encontrar a Seqüência Maximal de Nós à Esquerda

Dado um EE convexo A, quando a seqüência mais à esquerda maximal de $\mathcal{T}(A)$, digamos $[T_0, T_1, T_2, \cdots, T_j]$, é encontrada, então, pelo Teorema 7.15, $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição ótima de A. Mostraremos que a complexidade para encontrar a seqüência mais à esquerda maximal de $\mathcal{T}(A)$ é $O(n^4)$, onde n é o valor máximo entre as coordenadas do vetor retangular $\rho(A)$.

Seja k um inteiro positivo. Sejam A e Y dois EE's tais que Y é um nó de $\mathcal{T}(A)$ no nível k. Seja $\mathsf{Path}_A[\{o\}, Y] = [B_1, B_2, \cdots, B_k]$. Para verificar se Y é um invariante de A, temos que comparar se $A = (A \ominus Y) \oplus Y = (\cdots((A \ominus B_1) \ominus B_2 \ominus \cdots \ominus B_j) \oplus B_1) \oplus B_2 \oplus \cdots \oplus B_k)$. A complexidade para computar $A \ominus B_i$ ou $A \oplus B_i$ é linear com relação ao número de pontos de A, uma vez que B_i contém no máximo 9 pontos. Se n é o valor máximo entre as coordenadas do vetor retangular $\rho(A)$, então a complexidade de tempo para computar $A \ominus B_i$ ou $A \oplus B_i$ é $O(n^2)$. Assim, a complexidade para verificar se um nó Y no nível k é invariante de A é $O(k \cdot n^2)$.

Uma vez que o algoritmo apresentado na Seção 7.2 faz uma busca em profundidade, então a primeira seqüência de nós visitada pelo algoritmo até chegar a uma folha é a seqüência mais à esquerda maximal $[T_0, T_1, T_2, \cdots, T_j]$ de $\mathcal{T}(A)$. Se A é um EE convexo, então pelo, Teorema 7.15, $\mathsf{Path}_A[\{o\}, T_j]$ é uma decomposição ótima de A. Assim, a primeira solução encontrada pelo algoritmo proposto é sempre uma decomposição ótima de A.

Seja A um EE convexo. Pelo Teorema 4.18, $lower(A) = \lceil \max\{\rho_0(A), \rho_1(A)\}/2 \rceil = O(n)$, onde $n = \max\{\rho_0(A),\ \rho_1(A)\}$. Se a seqüência mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, T_2, \cdots, T_j]$, é maximal, então, pela Proposição 7.14, $j = lower(A)$. Logo, o tempo para o algoritmo encontrar a seqüência maximal mais à esquerda de $\mathcal{T}(A)$ é $O(1 \cdot n^2) + O(2 \cdot n^2) + \cdots + O(lower(A) \cdot n^2)$. Portanto, a complexidade para o algoritmo encontrar uma decomposição ótima de um EE convexo A é $O(lower(A)^2 \cdot n^2)$, ou seja, $O(n^4)$, uma vez que $lower(A) = O(n)$.

7.3.3 Algoritmo de Park e Chin

Park e Chin [35] desenvolveram uma extensão do algoritmo de Xu para encontrar decomposição de EE's simplesmente conexos, onde todos elementos na decomposição são também simplesmente conexos. Vimos no Capítulo 6 que existem famílias infinitas de EE's simplesmente conexos que são decomponíveis mas não têm decomposição segundo a definição de Park e Chin.

Embora a complexidade do algoritmo proposto seja $O(2^m)$, onde m é o número de elementos na seqüência de invariantes, ele tem uma vantagem em relação ao algoritmo de Park e Chin, uma vez que ele é capaz de decompor qualquer EE decomponível, incluindo os EE's nas famílias $\mathcal{X}$, $\mathcal{Y}$ e $\mathcal{Z}$ definidas no Capítulo 6.

7.4 Resultados Experimentais

Nesta seção, apresentaremos alguns resultados experimentais de nosso algoritmo para encontrar uma decomposição ótima para diferentes tipos de EE's: discos euclidianos (veja definição abaixo), EE's convexos (veja um exemplo na Figura 7.5.a), EE's conexos, decomponíveis, que contém buracos (veja um exemplo na Figura 7.5.b), EE's conexos, decomponíveis, que não contém buracos (veja um exemplo na Figura 7.5.c), EE's não conexos, decomponíveis, que contém buracos (veja um exemplo na Figura 7.5.d) e EE's não conexos, decomponíveis, que não contém buracos (veja um exemplo na Figura 7.5.e). Estes experimentos foram realizados em uma máquina "Sun Ultra Enterprise 3000". O tempo de processamento da CPU está medido em horas (h), minutos (m) e segundos (s).

O *disco euclidiano* de raio $r > 0$, centrado na origem, é o EE definido por $D(r) = \{(x, y) \in \mathbb{Z}^2 : x^2 + y^2 \leq r^2\}$ (veja a Figura 7.6 para alguns exemplos). Note que esta não é a única definição de EE's circulares discretos. Um método para se obter alguns tipos de EE's circulares discretos e suas respectivas decomposições pode ser encontrado em [48].

Dividiremos esta seção em três partes. Na primeira, mostramos alguns resultados para discos euclidianos; na segunda, para EE's convexos; e na terceira, para EE's decom-

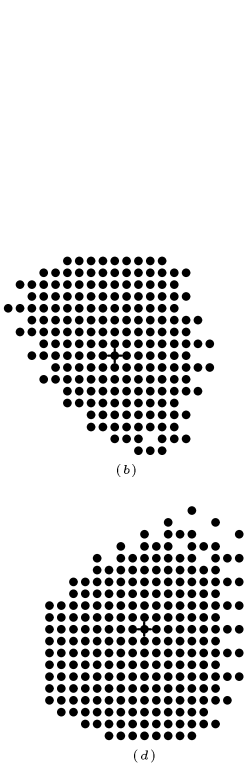

Figura 7.5: (a) Um EE convexo. (b) Um EE conexo, decomponível, que contém um buraco. (c) Um EE conexo, decomponível, que não contém buracos. (d) Um EE não conexo, decomponível, que contém alguns buracos. (e) Um EE não conexo, decomponível, que não contém buracos.

poníveis, conexos ou não, que contém buracos ou não. Nas tabelas desta seção, usaremos a seguinte notação:

NSI = número de elementos na seqüência de invariantes.

NFS = número de elementos na primeira solução encontrada.

NOS = número de elementos na solução ótima encontrada.

TND = tempo para detectar a não decomposabilidade de um EE.

TFS = tempo para detectar a primeira solução.

TOS = tempo para detectar a solução ótima.

7.4.1 Discos Euclidianos

Os discos $D(2)$ e $D(4)$ são decomponíveis, enquanto os discos $D(3)$ e $D(5)$ a $D(50)$ não têm decomposição. Os tempos para detectar a não decomposabilidade dos discos de $D(5)$ a $D(50)$ foram menores que 40 segundos. Os tempos para detectar a primeira solução

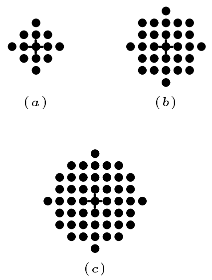

Figura 7.6: $(a) - (c)$ Discos euclidianos de raios 2, 3 e 4.

raio	NSI	NFS=NOS	TFS=TOS
2	5	2	0.5s
4	16	4	0.7s

Tabela 7.1: Discos decomponíveis.

(que também foi a ótima) dos discos $D(2)$ e $D(4)$ estão apresentados na Tabela 7.1. A Tabela 7.2 mostra o tempo para detectar a não decomposibilidade de alguns discos de raios entre 5 e 50. Somente para os discos $D(15)$, $D(20)$ e $D(25)$, o primeiro teste do algoritmo (isto é, a condição necessário para a existência de uma decomposição) foi satisfeito. Assim, isto explica o fato de que o tempo para detectar a não decomposibilidade de $D(25)$ é maior que $D(30)$.

7.4.2 Elementos Estruturantes Convexos

O algoritmo foi aplicado para encontrar uma decomposição ótima de aproximadamente 250 EE's convexos gerados aleatoriamente. Como foi mostrado no Teorema 7.15, toda primeira solução encontrada pelo algoritmo é uma solução ótima. A Tabela 7.3 apresenta o tempo médio para detectar a solução ótima para EE's convexos que são subconjuntos dos quadrados 20×20, 40×40, 56×56, 72×72 e 88×88. Por exemplo, o EE convexo apresentado na Figura 7.5.a é um subconjunto do quadrado 20×20. Observamos que o tempo para encontrar a decomposição ótima de EE's convexos é muito pequeno, mesmo para EE's grandes. A Tabela 7.4 apresenta o tempo para detectar a solução ótima de alguns EE's.

A título de comparação com o algoritmo desenvolvido no Capítulo 5, mostramos na Tabela 7.5 contendo os tempos que o algoritmo proposto levou para encontrar uma decomposição ótima dos mesmos EE's indicados na Tabela 5.1. Para manter compatibilidade com o experimento apresentado no Capítulo 5, este particular experimento foi executado em uma máquina com processador Pentium III, 450 MHz.

Vemos, na Tabela 7.5, que o algoritmo DecCon é muito mais rápido do que o algoritmo proposto. Isto não é tanto devido ao fato de que a complexidade de DecCon seja

raio	NSI	TND
05	4	0.7s
10	22	1.0s
15	38	2.2s
20	54	11.1s
25	70	36.3s
30	70	2.5s
35	54	4.4s
40	70	4.9s
45	86	6.3s
50	102	7.7s

Tabela 7.2: Discos que não são decomponíveis.

Subconjunto do quadrado	TFS=TOS
20×20	5.5s
40×40	19.4s
56×56	47.2s
72×72	1m42.0s
88×88	3m21.0s

Tabela 7.3: Tempo médio para se detectar a solução ótima de EE's convexos.

$O(n^3)$ e do algoritmo proposto seja $O(n^4)$, onde n é o valor máximo entre as coordenadas do vetor retangular do EE de entrada. Este acréscimo de tempo é devido à construção da seqüência de invariantes pelo algoritmo proposto. Note que o algoritmo DecCon não constrói a seqüência de invariantes e a seqüência SeqQ (do Passo 1) é construída previamente antes da execução do algoritmo. Para terminar esta subseção, observamos que as soluções obtidas pelo algoritmo proposto e pelo algoritmo DecCon foram as mesmas.

7.4.3 Elementos Estruturantes Decomponíveis, Conexos e não Conexos, com e sem Buracos

Nesta subseção, usaremos a seguinte notação para denotar os EE's:

DCH = EE's decomponíveis e conexos que contém buracos

DC = EE's decomponíveis e convexos que não contém buracos.

DDH = EE's decomponíveis e não conexos que contém buracos.

DD = EE's decomponíveis e não conexos que não contém buracos.

Aplicamos o algoritmo para encontrar a primeira solução (respectivamente, solução ótima) em aproximadamente 400 (respectivamente, 200) EE's decomponíveis, conexos e não conexos, com e sem buracos. Estes EE's foram gerados aleatoriamente. Os EE's aprensetados nas Figuras de 7.5.b a 7.5.e são alguns exemplos desses EE's.

A Tabela 7.6 apresenta o tempo médio para detectar a primeira solução de EE's que são subconjuntos do quadrado 20×20 e 40×40. Por exemplo, os EE's apresentados nas

EE A	$\rho(A)$	NSI	NFS=NOS	TFS=TOS
01	(17, 17)	448	9	5.5s
02	(17, 19)	531	10	7.7s
03	(31, 34)	844	17	23.3s
04	(32, 36)	918	18	22.0s
05	(48, 48)	1637	24	50.0s
06	(52, 55)	1619	28	57.0s
07	(65, 68)	2114	34	2m7.0s
08	(69, 68)	2217	35	2m30.0s
09	(85, 82)	2586	43	3m55.0s
10	(88, 84)	2802	44	4m32.0s

Tabela 7.4: Tempo para se detectar a solução ótima de alguns EE's convexos.

	Tempo Médio para encontrar uma solução ótima		
Subconjuntos do retângulo	Algoritmo do Xu	Algoritmo DecCon	Algoritmo proposto
20×21	0.22s	0.23s	1.33s
38×40	0.23s	0.27s	3.25s
56×57	0.23s	0.26s	8.16s
73×73	0.26s	0.28s	17.48s
89×89	0.28s	0.28s	31.70s
109×108	0.25s	0.33s	67.95s
124×124	0.32s	0.54s	2m1.09s
141×144	0.29s	0.42s	3m15.36s
160×162	0.31s	0.47s	6m0.87s
176×179	0.35s	0.56s	8m17.34s
192×193	0.37s	0.63s	11m20.19s
214×210	0.41s	0.75s	18m20.30s
229×227	0.45s	0.86s	24m23.05s
246×247	0.48s	0.97s	34m30.58s

Tabela 7.5: Tempo médio para encontrar uma decomposição ótima de EE's convexos.

Figuras de 7.5.b a 7.5.e são subconjuntos do quadrado 20×20. Na maioria dos casos, a primeira solução encontrada foi também a solução ótima. Além disso, a diferença entre o número de elementos na primeira solução e o limite inferior de uma decomposição foi no máximo dois. Portanto, na maioria das vezes, a primeira solução esteve muito perto da solução ótima.

Na Tabela 7.7, apresentamos o tempo para detectar a primeira solução e a solução ótima de alguns EE's que são subconjuntos do quadrado 20×20 e que o limite inferior não foi igual ao número de elementos na primeira solução. Nesta tabela, observe que a diferença entre o número de elementos na primeira solução e o limite inferior é normalmente pequeno e, na maioria dos casos, a primeira solução é a solução ótima. Uma vez que a complexidade de tempo para detectar a solução ótima, no pior caso, é exponencial, em aplicações práticas, uma boa heurística é parar quando a primeira solução é encontrada.

Embora os resultados apresentados nas Tabelas 7.6 e 7.7 sejam bons, o tempo para encontrar a primeira solução aumentou exponencialmente com o acréscimo do tamanho do EE. Isto aconteceu pois as três estratégias de podas usadas no algoritmo não foram

	20×20	40×40
DCH	3.266s	13m41.0s
DC	2.876s	22.2s
DDH	1.682s	34.0s
DD	1.506s	18m27.0s

Tabela 7.6: Tempo médio para detectar a primeira solução de alguns EE's.

EE A	$\rho(A)$	NSI	limite inferior	NFS	TFS	NOS	TOS
	(18,16)	17	9	10	1.2s	10	3.0s
	(16,16)	17	8	9	1.0s	9	4.1s
DDH	(18,18)	38	9	10	1.8s	10	29.4s
	(18,15)	21	9	10	1.1s	10	6.0s
	(14,18)	10	9	10	1.0s	10	1.3s
	(16,16)	19	8	9	1.1s	9	7.5s
DD	(18,17)	29	9	10	1.2s	10	43.5s
	(18,14)	36	9	10	1.5s	10	1m54.0s
	(18,16)	20	9	10	1.3s	10	7.7s
	(18,16)	24	9	10	1.2s	10	11.6s
	(16,16)	41	8	9	3.5s	9	2m39.0s
DCH	(18,18)	30	9	10	1.2s	10	57.2s
	(16,18)	19	9	10	1.0s	10	2.5s
	(16,18)	30	9	10	1.8s	10	18.0s
	(16,17)	60	9	10	4.1s	9	47.0s
DC	(14,18)	43	9	10	5.3s	10	2m41.0s
	(16,15)	31	8	9	1.6s	9	27.0s
	(18,18)	41	9	10	1.6s	10	8m20.3s

Tabela 7.7: Tempo para detectar a primeira solução e a solução ótima de alguns EE's.

capazes de evitar um número considerável de nós não viáveis na árvore de decomposição. A Tabela 7.8 apresenta o tempo para detectar a primeira solução de alguns EE's que são maiores que os EE's apresentados na Tabela 7.7.

7.5 Conclusão

A mudança ou transformação de estrutura de representação de operadores para melhorar o desempelho de suas implementações é um passo fundamental no processo de programação automática de MaqM's [11]. Neste capítulo, estudamos um aspecto particular deste problema: a decomposição seqüencial de erosões (respectivamente, dilatações).

Um algoritmo geral para prova automática de que uma erosão (respectivamente, dilatação) tenha uma decomposição seqüencial ou não foi apresentado. A prova da existência é construtiva e uma solução ótima é exibida. Este algoritmo é baseado em uma busca que usa o método "branch and bound", com estratégias de podas baseadas em propriedades geométricas e algébricas deduzidas formalmente.

O algoritmo proposto não é eficiente para todos os casos, mas generaliza importantes resultados clássicos (Zhuang and Haralick, Xu, Park e Chin). Uma análise teórica do algoritmo, juntamente com alguns resultados experimentais, ilustra estes fatos. O algoritmo

EE A	$\rho(A)$	NSI	TFS
DDH	(62,75)	114	67h38m56.0s
DD	(52,56)	91	5h25m30.0s
DCH	(50,54)	137	23h24m21.0s
DC	(50,58)	521	Pelo menos 201h

Tabela 7.8: Tempo para detectar a primeira solução de alguns EE's.

combinatorial apresentado pode ser melhorado se novos limites ou estratégias de podas forem descobertos.

7.6 Apêndice do Capítulo 7

Neste apêndice, vamos provar a Proposição 7.13. Dividiremos esta prova em duas partes. Na primeira, mostraremos alguns resultados intermediários necessários para a prova e, na segunda parte, provaremos a Proposição 7.13.

7.6.1 Resultados Necessários para a Prova da Proposição 7.13

Dado um EE A, para mostrar que uma seqüência maximal de nós mais à esquerda de $\mathcal{T}(A)$, digamos $[T_0, T_1, \cdots, T_j]$, é uma seqüência elemento retangular de A, precisamos de alguns resultados preliminares, dados nos Lemas de 7.16 a 7.18.

Lema 7.16 *Seja A um EE convexo. Seja $[T_0, T_1, \cdots, T_j]$ uma seqüência de nós mais à esquerda de $\mathcal{T}(A)$. Seja $\mathsf{Path}_A[\{o\}, T_j] = [D_1, D_2, \cdots, D_j]$ e seja B um elemento de $\mathsf{SeqInv}[A]$. Se, para algum $k \in \{1, 2, \cdots, j\}$, $T_{k-1} \oplus B$ é um invariante de A, então $\rho_0(D_k) + \rho_1(D_k) \geq \rho_0(B) + \rho_1(B)$.*

Prova: Uma vez que B é um elemento de $\mathsf{SeqInv}[A]$, a seqüência $[T_0, T_1, \cdots, T_{k-1}]$ é uma seqüência de nós mais à esquerda de $\mathcal{T}(A)$ e $T_{k-1} \oplus B$ é um invariante de A, então, pelo Lema 7.9, B é um elemento de $\mathsf{Arestas}_A[T_{k-1}]$. Como D_k é o primeiro elemento de $\mathsf{Arestas}_A[T_{k-1}]$ tal que $T_{k-1} \oplus D_k$ é invariante de A, então B aparece depois de D_k em $\mathsf{Arestas}_A[T_{k-1}]$. Assim, pela ordem escolhida para construir a seqüência $\mathsf{SeqInv}[A]$, $\rho_0(D_k) + \rho_1(D_k) \geq \rho_0(B) + \rho_1(B)$. ■

Lema 7.17 *Seja A um EE convexo. Seja $[D_1, D_2, \cdots, D_j]$ uma subseqüência de $\mathsf{SeqInv}[A]$ com $j \geq 2$ tal que $D_1 \oplus D_2 \oplus \cdots \oplus D_j$ é um invariante de A. Sejam D_i e D_k dois elementos de $[D_1, D_2, \cdots, D_j]$ tais que $i < k$. Se T é um EE tal que $|T| \geq 2$, $\rho(T) \leq (2, 2)$ e T é um invariante de $D_i \oplus D_k$ ou $C(D_i) \oplus C(D_k)$, então existe um B em $\mathsf{SeqInv}[A]$ tal que $\rho(B) = \rho(T)$ e $D_1 \oplus D_2 \oplus \cdots \oplus D_{i-1} \oplus B$ é im invariante de A.*

Prova: Seja $X = D_1 \oplus D_2 \oplus \cdots \oplus D_{i-1}$. Seja Y o resultado da adição de Minkowski de todos elementos em $[D_i, D_{i+1}, \cdots, D_j]$ com exceção de D_i e D_k. Assim, $Z = D_1 \oplus D_2 \oplus \cdots \oplus D_j = Y \oplus X \oplus D_i \oplus D_k$.

Uma vez que $Z = D_1 \oplus D_2 \oplus \cdots \oplus D_j$ é invariante de A, pela Proposição 3.14, existe um EE R tal que $A = R \oplus Z = R \oplus (Y \oplus X \oplus D_i \oplus D_k) = (R \oplus Y) \oplus (X \oplus D_i \oplus D_k) = S \oplus (X \oplus D_i \oplus D_k)$, onde $S = R \oplus Y$. Dessa forma, pela Proposição 3.15, $D_i \oplus D_k$ é um invariante de A.

Como A é um EE convexo e $A = S \oplus X \oplus D_i \oplus D_k$, então, pela Proposição 5.4, $A = S \oplus X \oplus C(D_i) \oplus D_k = S \oplus X \oplus C(D_i) \oplus C(D_k)$. Dessa forma, pela Proposição 3.15, $C(D_i) \oplus C(D_k)$ é um invariante de A.

Uma vez que T é um invariante de $D_i \oplus D_k$ ou $C(D_i) \oplus C(D_k)$; e $D_i \oplus D_k$ e $C(D_i) \oplus C(D_k)$ são invariantes de A, então, pela Proposição 5.2, T é um invariante de A.

Como que $\rho(T) \leq (2, 2)$ e T é um invariante de A, então existe um elemento B em $\mathsf{SeqInv}[A]$ tal que $B \equiv T$. Claramente, $\rho(B) = \rho(T)$. Agora, falta demonstrar que $X \oplus B$ é um invariante de A.

Uma vez que T é um invariante de $D_i \oplus D_k$ ou $C(D_i) \oplus C(D_k)$, então, claramente, B também é um invariante de $D_i \oplus D_k$ ou $C(D_i) \oplus C(D_k)$.

Se B é um invariante de $D_i \oplus D_k$, então, pela Proposição 3.14, existe um EE D tal que $B \oplus D = D_i \oplus D_k$. Assim, como $A = S \oplus X \oplus D_i \oplus D_k = S \oplus X \oplus (D_i \oplus D_k) = S \oplus (X \oplus B \oplus D) = S \oplus (X \oplus B) \oplus D$, então, pela Proposição 3.15, $X \oplus B$ é um invariante de A.

Analogamente, se B é um invariante de $C(D_i) \oplus C(D_k)$, então $X \oplus B$ é um invariante de A. ∎

Lema 7.18 *Seja A um EE convexo. Seja $[D_1, D_2, \cdots, D_j]$ uma subseqüência de* $\mathsf{SeqInv}[A]$ *com $j \geq 2$ tal que $D_1 \oplus D_2 \oplus \cdots \oplus D_j$ é um invariante de A. Se existem D_i e D_k em $[D_1, D_2, \cdots, D_j]$ tais que $\rho(D_i) = (2, 1)$ e $\rho(D_k) = (1, 2)$, então existe B em* $\mathsf{SeqInv}[A]$ *tal que $\rho(B) = (2, 2)$ e $D_1 \oplus D_2 \oplus \cdots \oplus D_{\ell-1} \oplus B$ é um invariante de A, onde $\ell = \min\{i, k\}$.*

Prova: Sejam $\mathcal{R}$, $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{S}$, $\mathcal{S}_1$ e $\mathcal{S}_2$ os conjuntos definidos na Proposição 5.24, cujos elementos estão mostrados nas Figuras 5.3 e 5.4. Da mesma forma que foi feito na Proposição 5.24, suponha que $C(D_i) \in \mathcal{R}$ e $C(D_k) \in \mathcal{S}$. Assim, uma vez que $\{\mathcal{R}_1, \mathcal{R}_2\}$ e $\{\mathcal{S}_1, \mathcal{S}_2\}$ são partições, respectivamente, de $\mathcal{R}$ e $\mathcal{S}$, temos dois casos para considerar: (1) $C(D_i) \in \mathcal{R}_1$ ou $C(D_k) \in \mathcal{S}_1$; e (2) $C(D_i) \in \mathcal{R}_2$ e $C(D_i) \in \mathcal{S}_2$.

<u>*Caso 1*</u>: $C(D_i) \in \mathcal{R}_1$ ou $C(D_k) \in \mathcal{S}_1$.

Consideraremos que $C(D_i) \in \mathcal{R}_1$ (a prova para $C(D_k) \in \mathcal{S}_1$ pode ser feita de maneira similar).

Como indicado na Figura 5.5, se $C(D_i) \in \mathcal{R}_1$, então existem dois EE's convexos C_1 e C_2 tais que $C(D_i) = C_1 \oplus C_2$, $\rho(C_1) = (1,1)$ e $\rho(C_2) = (1,0)$. Dessa forma, $C(D_i) \oplus C(D_k) = (C_1 \oplus C_2) \oplus C(D_k) = C_1 \oplus (C_2 \oplus C(D_k)) = C_1 \oplus T$, onde $T = C_2 \oplus C(D_k)$. Assim, uma vez que T é um invariante de $C(D_i) \oplus C(D_k)$ (pela Proposição 3.15) e $\rho(T) = \rho(C_2) + \rho(C(D_k)) = (2,2)$ (pela Proposição 4.17), então, pelo Lema 7.17, existe um elemento B em $\mathsf{SeqInv}[A]$ tal que $\rho(B) = \rho(T) = (2,2)$ e $D_1 \oplus D_2 \oplus \cdots \oplus D_{i-1} \oplus B$ é um invariante de A (se $i < k$) ou $D_1 \oplus D_2 \oplus \cdots \oplus D_{k-1} \oplus B$ é um invariante de A (se $k < i$).

Caso 2: $C(D_i) \in \mathcal{R}_2$ e $C(D_i) \in \mathcal{S}_2$.

Neste caso, a Figura 5.7 apresenta todas possibilidades de $C(D_i) \oplus C(D_k)$. Assim, como indicado nesta figura, existem dois EE's convexos C_1 e C_2 tais que $C(D_i) \oplus C(D_k) = C_1 \oplus C_2$, $\rho(C_1) = (2,2)$ e $\rho(C_2) = (1,1)$. Assim, pela Proposição 3.15, C_1 é um invariante de $C(D_i) \oplus C(D_k)$. Dessa forma, pelo Lema 7.17, existe B em $\mathsf{SeqInv}[A]$ tal que $\rho(B) = \rho(C_1) = (2,2)$ e $D_1 \oplus D_2 \oplus \cdots \oplus D_{i-1} \oplus B$ é um invariante de A (se $i < k$) ou $D_1 \oplus D_2 \oplus \cdots \oplus D_{k-1} \oplus B$ é um invariante de A (se $k < i$).

Portanto, existe B em $\mathsf{SeqInv}[A]$ tal que $\rho(B) = (2,2)$ e $D_1 \oplus D_2 \oplus \cdots \oplus D_{\ell-1} \oplus B$ é im invariante de A, onde $\ell = \min\{i,k\}$. ■

Agora, com estes resultados dados acima, podemos mostrar que, dado um EE convexo A, se $[T_0, T_1, \cdots, T_j]$ é uma seqüência maximal de nós mais à esquerda de $\mathcal{T}(A)$, então $[T_0, T_1, \cdots, T_j]$ é uma seqüência elemento retangular de A.

7.6.2 Prova da Proposição 7.13

Seja $\mathsf{Path}_A[\{o\}, T_j] = [D_1, D_2, \cdots, D_j]$. Obviamente, por construção, $\mathsf{Path}_A[\{o\}, T_j]$ é a seqüência geradora de $[T_0, T_1, \cdots, T_j]$. Assim, para provar que a seqüência $[T_0, T_1, \cdots, T_j]$ é uma seqüência elemento retangular de A, basta mostrar que sua seqüência geradora $\mathsf{Path}_A[\{o\}, T_j]$ é uma seqüência retangular. Dessa forma, temos que provar que as Restrições da definição de seqüência retangular são válidas para a seqüência $\mathsf{Path}_A[\{o\}, T_j]$.

Prova que as Restrições de (a) a (f) são válidas:

Suponha, por absurdo, que uma das Restrições de (a) a (f) da definição de seqüência retangular não seja válida para a seqüência $\mathsf{Path}_A[\{o\}, T_j]$. Assim, existem dois elementos D_i e D_k, com $i < k$, na seqüência $\mathsf{Path}_A[\{o\}, T_j] = [D_1, D_2, \cdots, D_j]$ tais que $\rho(D_i \oplus D_k) = \rho(D_i) + \rho(D_k) \leq (2,2)$. Seja $T = D_i \oplus D_k$. Claramente, T é um invariante de $D_i \oplus D_k$. Além disso, como D_i e D_k são elementos de $\mathsf{SeqInv}[A]$, então $|T| \geq 2$ e $\rho_0(D_k) + \rho_1(D_k) > 0$.

Uma vez que $\rho(T) = \rho(D_i) + \rho(D_k) \leq (2,2)$, $|T| \geq 2$ e T é um invariante de $D_i \oplus D_k$, então, pelo Lema 7.17, existe um elemento B em $\mathsf{SeqInv}[A]$ tal que $\rho(B) = \rho(T)$ e $D_1 \oplus D_2 \oplus \cdots \oplus D_{i-1} \oplus B = T_{i-1} \oplus B$ é invariante de A. Como $i \in \{1, 2, \cdots, j\}$, então, pelo Lema 7.16, $\rho_0(D_i) + \rho_1(D_i) > \rho_0(B) + \rho_1(B)$.

Uma vez que $\rho(B) = \rho(T) = \rho(D_i) + \rho(D_k)$, então

$$\begin{aligned}\rho_0(D_i) + \rho_1(D_i) \geq \rho_0(B) + \rho_1(B) &= (\rho_0(D_i) + \rho_0(D_k)) + (\rho_1(D_i) + \rho_1(D_k)) \\ &= (\rho_0(D_i) + \rho_1(D_i)) + (\rho_0(D_k) + \rho_1(D_k)).\end{aligned}$$

Dessa forma, $\rho_0(D_k) + \rho_1(D_k) \leq 0$. Mas isto é um contradição, pois $\rho_0(D_k) + \rho_1(D_k) > 0$. Portanto, todas as Restrições de (a) a (f), são válidas para $\mathsf{Path}_A[\{o\}, T_j]$.

<u>Prova que as Restrições de (g) a (j) são válidas:</u>

Suponha por absurdo que uma das restrições, de (g) a (j), não seja válida. Dessa forma, $c_{2,0}(\mathsf{Path}_A[\{o\}, T_j]) > 0$ ou $c_{0,2}(\mathsf{Path}_A[\{o\}, T_j]) > 0$. Conseqüentemente, existe um elemento D_k em $\mathsf{Path}_A[\{o\}, T_j]$ tal que $\rho(D_k) = (2,0)$ ou $\rho(D_k) = (0,2)$. Consideraremos o caso $\rho(D_k) = (2,0)$. O outro caso $\rho(D_k) = (0,2)$ pode ser provado de uma maneira similar. Logo, pelas restrições (g) ou (h), $c_{1,1}(\mathsf{Path}_A[\{o\}, T_j]) > 0$ ou $c_{1,2}(\mathsf{Path}_A[\{o\}, T_j]) > 0$, e, portanto, existe um elemento D_i em $\mathsf{Path}_A[\{o\}, T_j]$ tal que $\rho(D_i) = (1,1)$ ou $\rho(D_i) = (1,2)$. Uma vez que $\rho_0(D_i) + \rho_1(D_i) \geq \rho_0(D_k) + \rho_1(D_k)$, pela ordem escolhida para construir $\mathsf{SeqInv}[A]$, podemos considerar que $i < k$ (o caso $i > k$ é analógo).

Uma vez que, $\rho(C(D_k)) = \rho(D_k) = (2,0)$ (pela definição de vetor retangular), então $C(D_k)$ é uma linha horizontal com três pontos consecutivos. Dessa forma, existem duas linhas horizontais com dois pontos consecutivos cada um, digamos L_1 e L_2, tais que $C(D_k) = L_1 \oplus L_2$. Claramente, $\rho(L_1) = \rho(L_2) = (1,0)$.

Assim, $C(D_i) \oplus C(D_k) = C(D_i) \oplus (L_1 \oplus L_2) = (C(D_i) \oplus L_1) \oplus L_2$. Dessa forma, pela Proposição 3.15, $T = C(D_i) \oplus L_1$ é um invariante de $C(D_i) \oplus C(D_k)$. Além disso, pela Proposição 4.17, $\rho(T) = \rho(C(D_i)) + \rho(L_1) \leq (2,2)$, uma vez que $\rho(D_i) = (1,1)$ ou $\rho(D_i) = (1,2)$.

Uma vez que, $\rho(T) \leq (2,2)$, B é um invariante de $C(D_i) \oplus C(D_k)$ e $i < k$, então, pelo Lema 7.17, existe um elemento B em $\mathsf{SeqInv}[A]$ tal que $\rho(B) = \rho(T)$ e $D_1 \oplus D_2 \oplus \cdots \oplus D_{i-1} \oplus B = T_{i-1} \oplus B$ é invariante de A. Dessa forma, uma vez que $i \in \{1, 2, \cdots, j\}$, então, pelo Lema 7.16, $\rho_0(D_i) + \rho_1(D_i) > \rho_0(B) + \rho_1(B)$.

Uma vez que $\rho(B) = \rho(T) = \rho(C(D_i)) + \rho(L_1)$ e $\rho(C(D_i)) = \rho(D_i)$, então

$$\begin{aligned}\rho_0(D_i) + \rho_1(D_i) \geq \rho_0(B) + \rho_1(B) &= (\rho_0(C(D_i)) + \rho_0(L_1)) + (\rho_1(C(D_i)) + \rho_1(L_1)) \\ &= (\rho_0(D_i) + \rho_0(L_1)) + (\rho_1(D_i) + \rho_1(L_1)) \\ &= (\rho_0(D_i) + \rho_1(D_i)) + (\rho_0(L_1) + \rho_1(L_1)).\end{aligned}$$

Assim, $\rho_0(L_1) + \rho_1(L_1) \leq 0$. Mas isto contradiz o fato de que $\rho(L_1) = (1,0)$. Logo, as Restrições de (g) a (j) são válidas para $\mathsf{Path}_A[\{o\}, T_j]$.

<u>Prova que a Restrição (k) é válida:</u>

Suponha por absurdo que a Restrição (k) não seja válida, ou seja, $c_{1,2}(\mathsf{Path}_A[\{o\}, T_j]) > 0$ e $c_{2,1}(\mathsf{Path}_A[\{o\}, T_j]) > 0$. Sejam D_i e D_k dois elementos em $\mathsf{Path}_A[\{o\}, T_j]$ tais que, $\rho(D_i) = (1,2)$ e $\rho(D_k) = (2,1)$. Sem perda de generalidade suponha que $i < k$.

Pelo Lema 7.18, existe B em $\mathsf{SeqInv}[A]$ tal que $\rho(B) = (2,2)$ e $D_1 \oplus D_2 \oplus \cdots \oplus D_{i-1} \oplus B =$

$T_{i-1} \oplus B$ é um invariante de A. Dessa forma, uma vez que $i \in \{1, 2, \cdots, j\}$, pelo Lema 7.16, $\rho_0(D_i) + \rho_1(D_i) > \rho_0(B) + \rho_1(B)$. Mas, isto contradiz o fato de que $\rho_0(D_i) + \rho_1(D_i) = 3$ e $\rho_0(B) + \rho_1(B) = 4$. Logo, $c_{1,2}(\mathsf{Path}_A[\{o\}, T_j]) = 0$ ou $c_{2,1}(\mathsf{Path}_A[\{o\}, T_j]) = 0$ e, portanto, $c_{2,1}(\mathsf{Path}_A[\{o\}, T_j]) \cdot c_{1,2}(\mathsf{Path}_A[\{o\}, T_j]) = 0$.