MAC5701 Tópicos em Ciência da Computação
Relatório de Estudos
Algoritmo paralelo para o problema da transversal mínima

Professora Responsável: Yoshiko Wakabayashi
Orientador: Siang Wun Song
Aluno: Rogério Seiji Ueda

24 de junho de 2002

Resumo

Este documento tem por finalidade apresentar um relatório dos estudos para MAC5701 - Tópicos em Ciência da Computação (2°semestre/2002), acerca dos algoritmos para solução do problema da transversal mínima (*hitting set problem*): uma heurística seqüêncial, a paralelização e implementação.

1 Tema

Algoritmos para o problema da transversal mínima (*hitting set problem*): uma heurística seqüêncial, a paralelização e implementação.

2 O problema da transversal mínima

Seja δ uma coleção finita de subconjuntos de um conjunto finito E. Um subconjunto T de E é uma **transversal** de δ se $T \cap S$ é não-vazio para cada S em δ. O **problema da transversal mínima** consiste no seguinte:

Problema MINTC(E, δ, c): Dados um conjunto finito E, uma coleção finita δ de subconjuntos de E e um custo c_e em $Q_{\geq}$ para cada e em E, encontrar uma transversal T de δ que minimize c(T) [2].

Uma formulação alternativa para o mesmo problema da transversal mínima é dado a seguir[7]:

Dados um conjunto finito E, uma coleção finita $\mathcal{S} = \{S_1, ..., S_w\}$ de subconjuntos de E, encontrar um conjunto $A \subseteq E$ de cardinalidade mínima, tal que $A \cap S_i \neq \emptyset$ para todo $i = 1, ..., w$.

O problema da transversal mínima, que é um problem NP-difícil, é equivalente ao conhecido problema da cobertura de conjuntos (*set cover problem*) [7]. Devido a essa equivalência, resultados para o problema da cobertura de conjuntos podem ser transformados em resultados para o problema da transversal mínima [8].

3 Algumas motivações

Algumas aplicações interessantes da solução deste problema, podem ser aplicados a diversas áreas:

3.1 Computação gráfica

3.2 Biologia molecular

Um dos objetivos da biologia molecular consiste no entendimento das funções dos genes e suas interações através da análise da expressão gênica [1] [4]. Acredita-se que as atividades celulares são definidos pelos genes expressos e portanto pela análise da expressão gênica seria possível determinar a relação entre os genes [6]. Um modelo de rede que pode representar o nível de expressão de cada gene é a rede booleana [2] [5].

[1] Expressão gênica é o processo no qual os genes produzem RNA e proteínas e exercem seus efeitos no fenótipo (Fenótipo é o conjunto de características observáveis de um organismo) [3].

[2] Uma rede booleana é representada como um grafo consistindo de N nós, que representam genes, numerados $a_n (0 \leq n < N)$, uma topologia de arestas direcionadas entre os nós, e uma função f_n para cada nó [1].

3.3 Prospecção de dados (*Datamining*)

Na análise comportamental dos compradores de uma rede de supermercados, por exemplo, a solução do problema da transversal mínima traz resultados interresantes nesta área, no tocante à análise do perfil de consumo dos compradores.

4 Um algoritmo seqüêncial

Um algoritmo aproximado seqüêncial para a solução deste problema é um algoritmo que segue uma estratégia "gulosa".

Seja E o conjunto de elementos e $\mathcal{S}$ a coleção de subconjuntos de E e sejam E' e $\mathcal{S}'$ conjuntos vazios inicialmente, o algoritmo deve executar os seguintes passos até que $\mathcal{S}' = \mathcal{S}$, [7, 9, 8]:

- Escolher um elemento e fora do conjunto $E \backslash E'$ que cobre o maior número de conjuntos na coleção $\mathcal{S} \backslash \mathcal{S}'$.

- Seja e o elemento escolhido no passo anterior e seja $S(e)$ a coleção de conjuntos em $\mathcal{S}$ cobertos por e:

$$E' \leftarrow E' \cup \{e\}$$
$$\mathcal{S}' \leftarrow \mathcal{S}' \cup S(e)$$

Seja $\mathcal{H}_d$ o d-ésimo número harmônico $\sum_{i=1}^{d} \frac{1}{i}$ e $S(e)$ o número de conjuntos na coleção $\mathcal{S}$ que são cobertos pelo elemento e [8].

Segundo Jha et al. [9], devido a equivalência entre o problema da cobertura de conjuntos e o problema da transversal mínima, o algoritmo guloso apresentado é uma ρ-aproximação polinomial para o *hitting set*, onde $\rho = \mathcal{H}(max_{e \in E}\{|S(e)|\})$, e segue ainda que a razão de aproximação é limitada por $\ln |\mathcal{S}| + 1$ [10].

5 Um exemplo seqüêncial

Um exemplo interessante será exposto para a área de prospecção de dados (*Datamining*).

Iniciando com uma proposta bastante genérica:

> *"Dados perfis de consumo dos clientes de um determinado supermercado, quero saber qual a relação entre os produtos comprados pelos clientes, isto é, desejo saber a interrelação dos produtos adquiridos pelos clientes."*

Seja tomado como exemplo a tabela de perfis a seguir, ilustrado na figura 1:

	Verdura Fresca	Comida Congelada	Frutas	Salgadinhos	Refrigerantes	Legumes Frescos
1	não comprou	não comprou	não comprou	não comprou	não comprou	não comprou
2	não comprou	não comprou	comprou	não comprou	não comprou	comprou
3	não comprou	comprou	não comprou	comprou	não comprou	não comprou
4	não comprou	comprou	comprou	comprou	comprou	comprou
5	comprou	não comprou	não comprou	não comprou	não comprou	comprou
6	comprou	não comprou	comprou	não comprou	comprou	comprou
7	comprou	comprou	não comprou	comprou	comprou	comprou
8	comprou	comprou	comprou	comprou	comprou	comprou

Figura 1: Exemplo de perfis de consumo.

Ao observar atentamente a tabela, uma pergunta mais específica pode vir a tona:

> *"Qual a relação entre 'Legumes Frescos' e os demais produtos adquiridos pelos clientes?"*

É possível responder a esta pergunta, solucionando o problema da transversal mínima que está implicito no problema apresentado.

A partir da figura 1, considere:

- *"Comprou"* codificado com valor "1"(um)
- *"Não comprou"* codificado com valor "0"(zero)

Esta nova interpretação, dos dados, resulta na seguinte tabela, representado pela figura 2:

	Verdura Fresca	Comida Congelada	Frutas	Salgadinhos	Refrigerantes	**Legumes Frescos**
	0	1	2	3	4	
1	0	0	0	0	0	**0**
2	0	0	1	0	0	**1**
3	0	1	0	1	0	**0**
4	0	1	1	1	1	**1**
5	1	0	0	0	0	**1**
6	1	0	1	0	1	**1**
7	1	1	0	1	1	**1**
8	1	1	1	1	1	**1**

Figura 2: Interpretando os dados de perfis de consumo.

A partir da nova tabela representado pela figura 2, observa-se que o conjunto E é formado por:

$$E = \{0, 1, 2, 3, 4\}$$

A partir do conjunto E e a partir da tabela representado pela figura 2, constrói-se os subconjuntos de E da seguinte forma:

1. Percorrer todas as linhas da tabela da figura 2.
2. Para cada linha da tabela da figura 2, fixa-se a linha atual e verifica-se se a variável "Legumes Frescos" teve seu valor alterado nas linhas subsequentes, isto é, se alterou-se de "0" para "1" ou de "1" para "0", a variável que se quer estudar (neste caso a variável "Legumes Frescos").
3. Se o valor alterou-se então é criado um subconjunto de E contendo todas as variáveis que também tiveram seu valor alterado paralelamente ao da variável "Legumes Frescos".

Ilustrando o procedimento anterior, no exemplo atual:

- Fixo a linha 1, o valor da variável "Legumes Frescos", neste caso tem valor "0".
- Fixado esta linha, para a linha seguinte, linha 2, percebe-se que o valor da variável "Legumes Frescos" alterou-se de "0" para "1".

Nesta situação, observa-se que somente a variável 2 (que representa "Frutas") teve seu valor alterado, as demais variáveis (0, 1, 3 e 4) mantiveram seus valores. Desta forma é criado um subconjunto com apenas um elemento: $\{2\}$.

- Ainda fixado a linha 1, para a linha 3, percebe-se que o valor da variável "Legumes Frescos"não foi alterado, isto é, o valor "0"da linha 1 manteve-se também "0"na linha 3 e portanto não é necessário fazer nada.
- Ainda fixado a linha 1, para a linha 4, percebe-se que o valor da variável "Legumes Frescos"alterou-se de "0"para "1". Nesta situação observa-se que as variáveis 1, 2, 3 e 4 também tiveram seus valores alterados. Desta forma é criado um sub conjunto com estes elementos, formando: $\{1,2,3,4\}$.
- Aplica-se esta lógica sucessivamente, fixando as linhas de 1 a 8 e repetindo-se o ciclo obtendo-se assim todos os subconjuntos de E.

Os subconjuntos construídos a partir desta lógica são as seguintes:

$$T_1 = \{2\}$$
$$T_2 = \{1,2,3,4\}$$
$$T_3 = \{0\}$$
$$T_4 = \{0,2,4\}$$
$$T_5 = \{0,1,3,4\}$$
$$T_6 = \{0,1,2,3,4\}$$
$$T_7 = \{1,2,3\}$$
$$T_8 = \{2,4\}$$
$$T_9 = \{0,1,3\}$$
$$T_{10} = \{0,1,2,4\}$$
$$T_{11} = \{0,4\}$$
$$T_{12} = \{0,2,4\}$$

A próxima etapa será a construção da estrutura de dados baseado nestes subconjuntos.

Observe a figura 3:

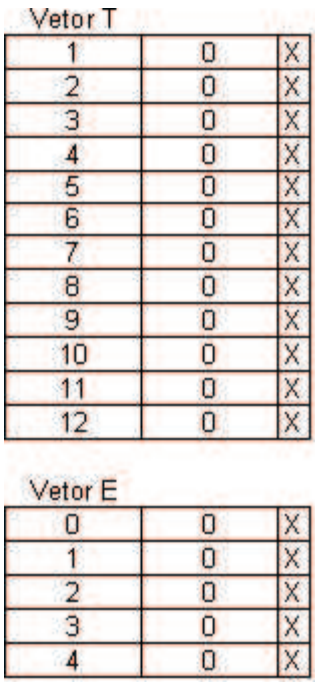

Figura 3: Estrutura de dados inicial, construído a partir dos subconjuntos calculados.

O vetor T representa os subconjuntos obtidos. Neste exemplo tem-se 12 subconjuntos numerados de 1 a 12.

Estrutura do vetor T:

- número do subconjunto (primeira coluna, inteiro que pode variar de 1 a 12).
- booleano que sinaliza se este subconjunto já foi coberto ou não (segunda coluna, o valor "1"significa que foi coberto e valor "0"caso contrário).
- ponteiro para uma lista ligada de elementos de E (terceira coluna, é o início de uma lista ligada, cujos nós são elementos do conjunto E, ou seja, elementos com valores de 0 a 4).

O vetor E representa os elementos do conjunto E, neste exemplo composto pelos elementos 0, 1, 2, 3 e 4.

Estrutura do vetor E:

- número do elemento no conjunto E (primeira coluna, inteiro que pode variar de 0 a 4).

- total de ocorrências deste elemento nos subconjuntos (segunda coluna, inteiro que pode variar de 1 a 12 (neste exemplo temos 12 conjuntos portanto o número máximo de vezes é 12)).
- ponteiro para uma lista ligada com os subconjuntos onde este elemento está presente (terceira coluna, é o início de uma lista ligada, cujos nós são elementos que representam os subconjuntos, ou seja, elementos com valores de 1 a 12).

A forma de se inicializar corretamente esta estrutura de dados consiste no seguinte:

1. Inicialmente as listas ligadas dos vetores: T e E estão vazias. O campo de contagem do vetor E e o *flag* de cobertura do vetor T, ambos devem iniciar com valores iguais a 0.
2. Para cada subconjunto T_i, onde $1 \leq i \leq 12$ (pois neste caso temos 12 subconjuntos) preencher:
   - No vetor T, para cada elemento do subconjunto T_i, atual, inserir os elementos na lista ligada, relativo ao subconjunto.
   - No vetor E, para cada elemento do subconjunto T_i, localizar a posição do elemento do subconjunto no vetor E, e inserir na lista ligada o valor i, incrementando também o contador do vetor, para cada elemento do subconjunto T_i.

O exemplo abaixo irá esclarecer o mecanismo de preenchimento da estrutura de dados:

Partindo da estrutura inicialmente vazia (figura 3) será introduzido nesta estrutura o primeiro subconjunto: (figura 4)

$$T_1 = \{2\}$$

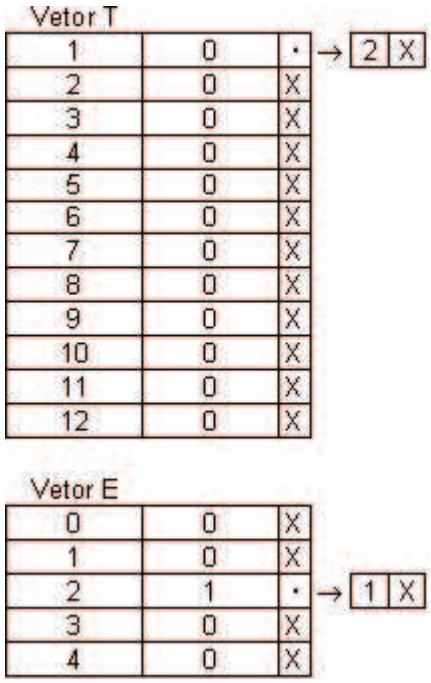

Figura 4: Após a inserção do subconjunto T_1.

Observe que no vetor T foi escolhido a posição 1 pois está se trabalhando com o subconjunto T_1. Sendo assim foi incluído na lista ligada o elemento 2, que corresponde ao elemento 2 do subconjunto ($T_1 = \{2\}$).

No vetor E foi escolhido a posição 2 pois o subconjunto T_1 possui o elemento 2, portanto o contador foi incrementado de 0 para 1 (pois foi inserido um elemento na lista ligada) e acrescentado o número 1 na lista ligada referente ao subconjunto T_1.

Passando para o próximo subconjunto:

$$T_2 = \{1, 2, 3, 4\}$$

E aplicando-se a mesma mecânica anterior, tem-se a figura 5

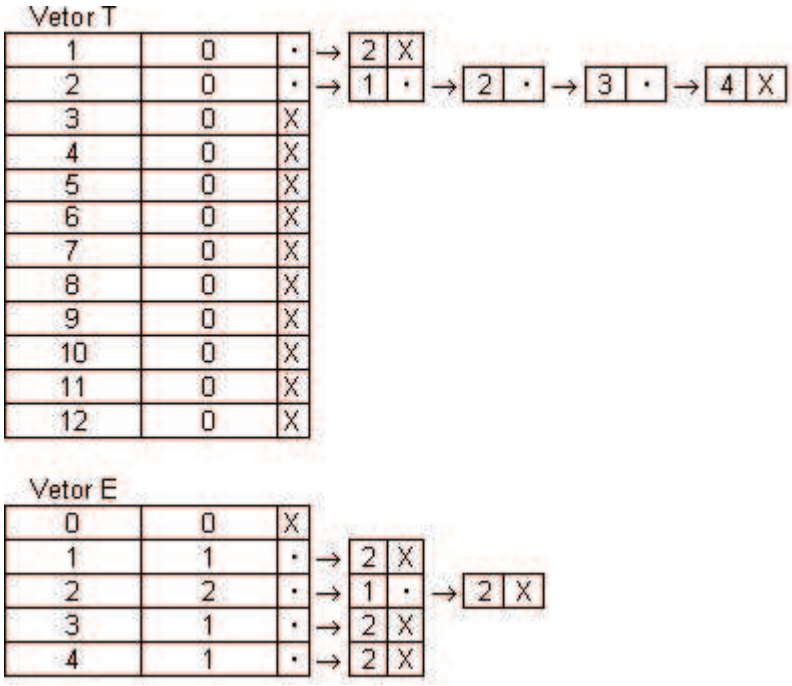

Figura 5: Após a inserção do subconjunto T_2.

Observe agora que no vetor T foi escolhido a posição 2 pois está se trabalhando com o subconjunto T_2. Sendo assim foi incluído na lista ligada desta posição os elementos:

- 1
- 2
- 3
- 4

que correspondem ao elementos do subconjunto $T_2 = \{1, 2, 3, 4\}$.

As listas ligadas dos elemntos do vetor E correspondentes aos elementos:

- 1
- 2
- 3

- 4

tiveram sua listas modificadas, isto é, para as posições do vetor 1, 2, 3 e 4 houve a inserção do elemento 2 ao final de suas respectivas listas (o valor 2 refere-se ao subconjunto T_2, com a qual está se trabalhando). Observa-se também que os respectivos contadores do vetor E foram incrementados de 1, devido ao crescimento da lista ligada.

Passando agora para o próximo subconjunto:

$$T_3 = \{0\}$$

E aplicando a lógica anterior, tem-se como resultado a seguinte estrutura de dados representado pela figura 6:

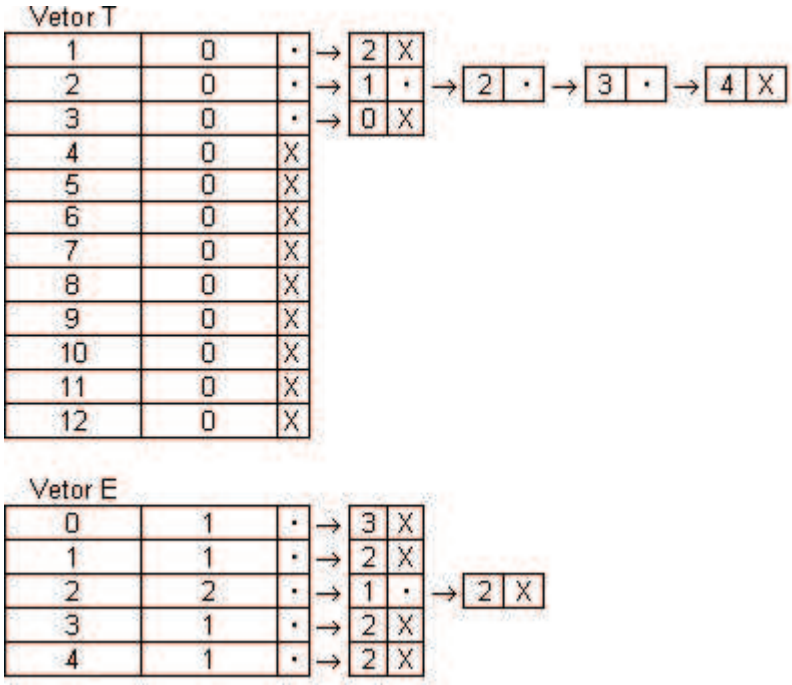

Figura 6: Após a inserção do subconjunto T_3.

Como o subconjunto envolvido é o $T_3 = \{0\}$, então a atualização no vetor T na posição 3 é feito inserindo-se na lista ligada os elementos do subconjunto, que neste caso corresponde apenas ao elemento 0. No vetor E, é feito alteração na posição 0 pois o elemento do subconjunto T_3 é o 0, e portanto é feito o incremento do contador de 0 para 1, e a inserção na lista ligada do número do subconjunto envolvido, neste caso 3 (T_3).

Aplicando-se este procedimento para todos os 12 subconjuntos, tem-se a estrutura de dados totalmente carregada e pronta para ser processada (figura 7):

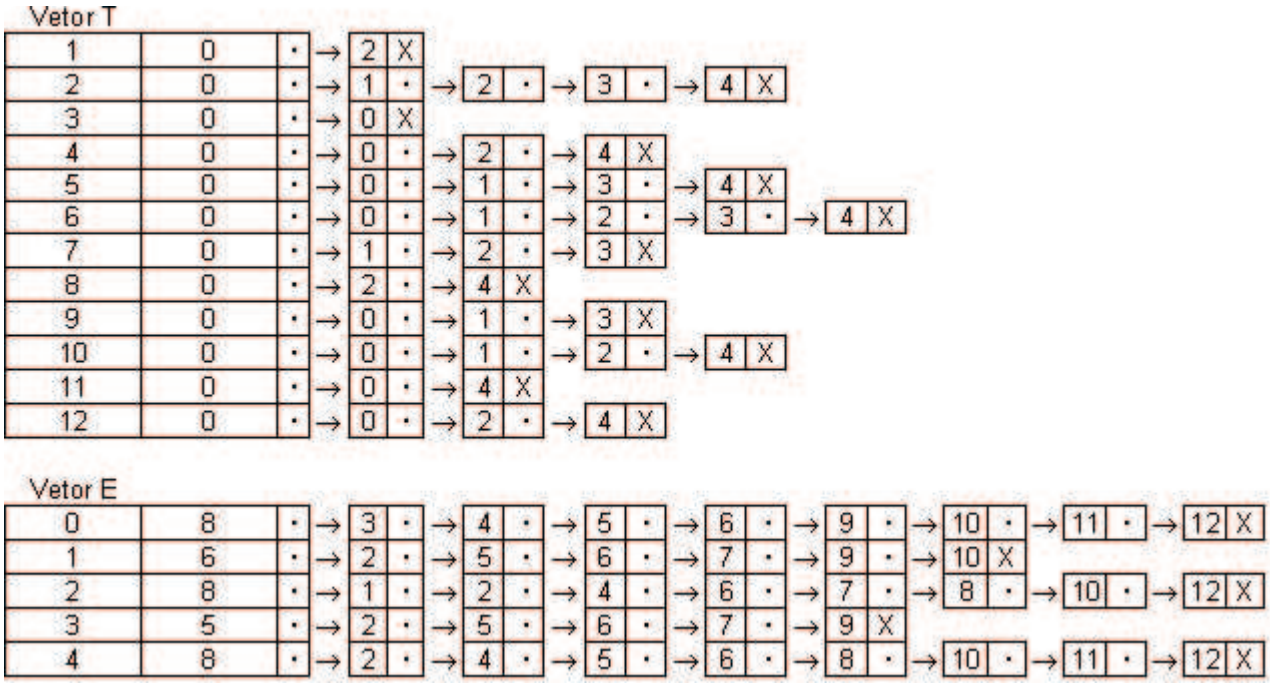

Figura 7: Após a inclusão de todos os 12 subconjuntos.

A partir deste ponto a estrutura de dados está preparada. Deve-se então aplicar o seguinte procedimento:

1. Fazer $E' = \varnothing$.
2. No vetor E, enquanto o max{"contadores do vetor E"} $\neq 0$ fazer para a lista ligada do vetor E de maior comprimento [3]:
   - Colocar este elemento e em E' ($E' = E' \cup \{e\}$.
   - Para cada elemento da lista ligada escolhida de E, marcar o campo *flag* como 1 para o correspondente elemento no vetor T, e, percorrendo a lista ligada de T (deste elemento), subtrair do contador do vetor E de um, para os elementos desta lista.
   - A cada elemento da lista ligada anterior, deve-se retirar o elemento da lista.

O exemplo a seguir irá esclarecer este mecanismo de escolha dos elementos.

[3] Neste caso o "maior comprimento da lista ligada"do vetor E significa o "maior contador do vetor E", pois estes contadores representam o número de vezes que cada elemento de E "aparece"nos subconjuntos.

A partir da figura 7, verifica-se que o maior contador do vetor E correponde ao elemento 0, portanto tem-se:

$$E' = \varnothing, \text{ inicialmente}$$

$$E' = E' \cup \{0\}, \text{ posteriomente}$$

com sua correspondente alteração na estrutura de dados representado pela seguinte seqüência:

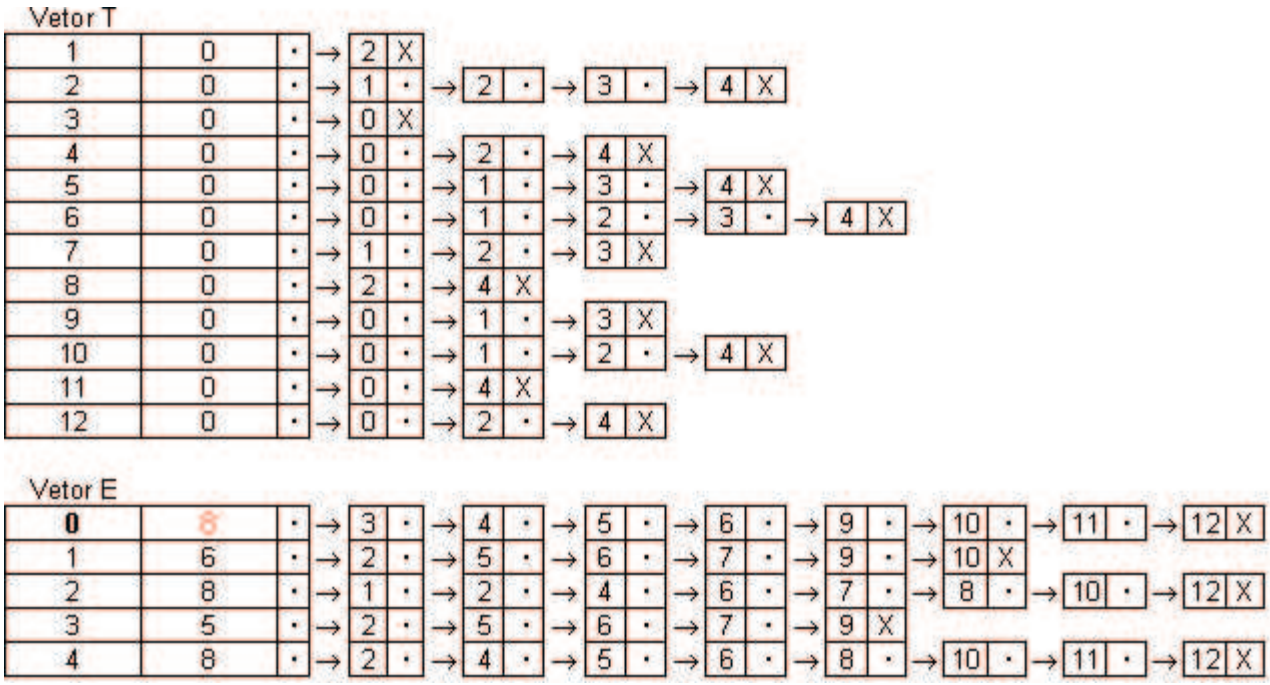

Figura 8: O elemento escolhido é 0, com 8 ocorrências.

Nesta etapa localiza-se o "maior contador"do vetor E (figura 8), neste caso corresponde ao elemento 0 do vetor E. Faz-se $E' = E' \cup \{0\}$ e então começa-se a percorrer a lista ligada deste elemento.

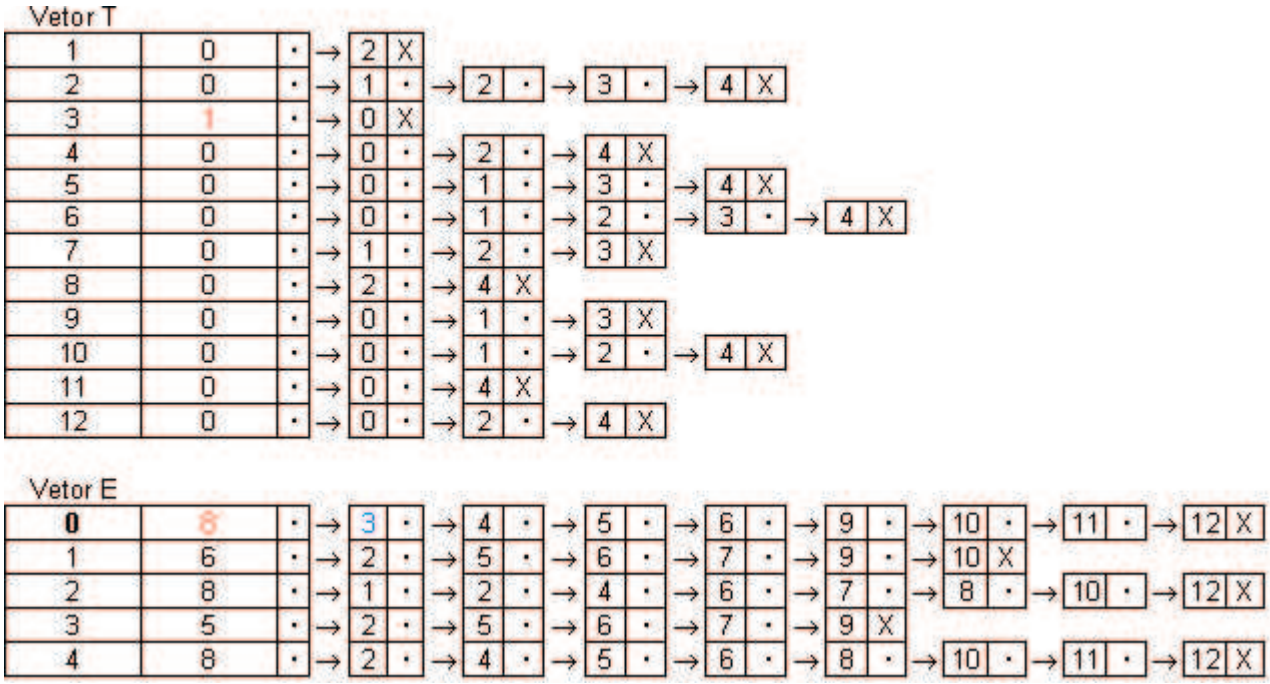

Figura 9: No vetor E, percorre-se a lista ligada de 0.

Para o primeiro elemento da lista ligada do vetor E, isto é, o elemento 3 faz-se o seguinte:

- Localiza-se o elemento 3 no vetor T.
- Marca-se no campo *flag* [4] atribuindo o valor 1 ao mesmo.
- Percorre-se a lista ligada do vetor T, neste caso na posição 3. Para cada elemento da lista ligada (aqui tem-se apenas o elemento 0 (figura 10)), fazer:
  1. Na posição 0 do vetor E, subtrair de 1 o contador $(8-1=7)$, conforme a figura 10.

[4] Este campo indica se o subconjunto já foi ou não "coberto"

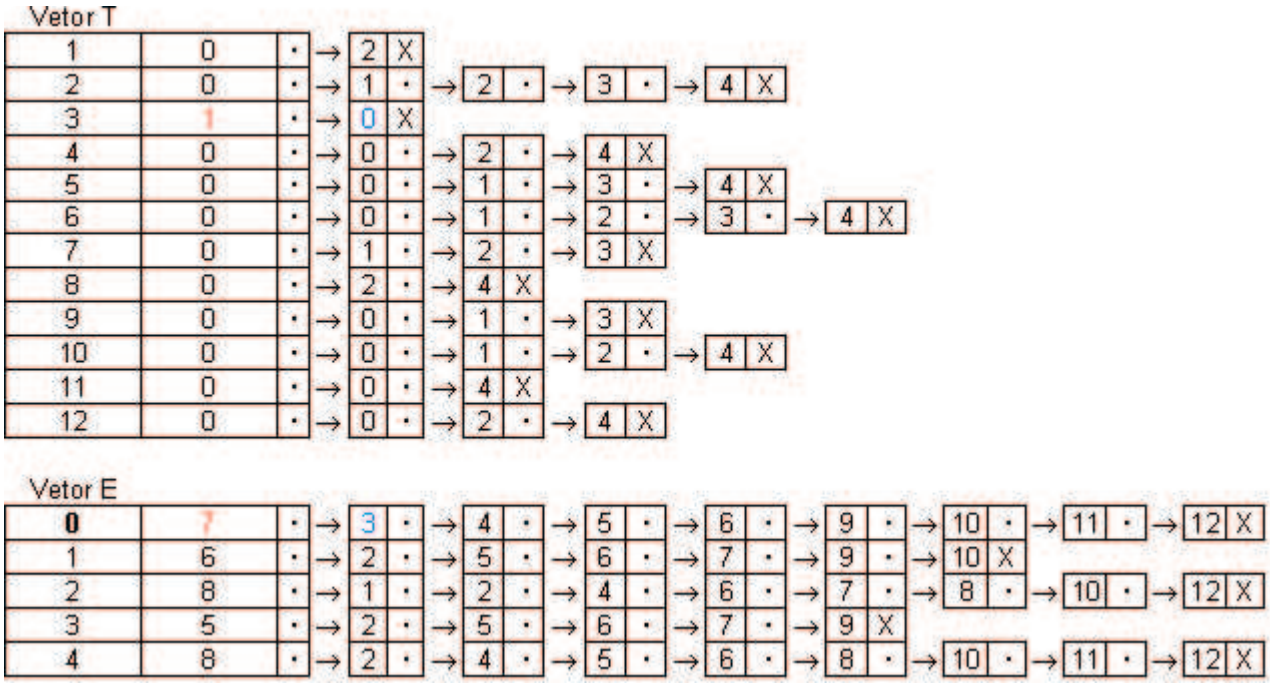

Figura 10: No vetor E, decrementa-se o contador do elemento 0, de 8 para 7.

Iterando a lógica descrita anteriormente para todos os elementos da lista ligada do elemento 0 do vetor E, isto é, os elementos:

- 3
- 4
- 5
- 6
- 9
- 10
- 11
- 12

Tem-se os seguintes passos na alteração da estrutura de dados:

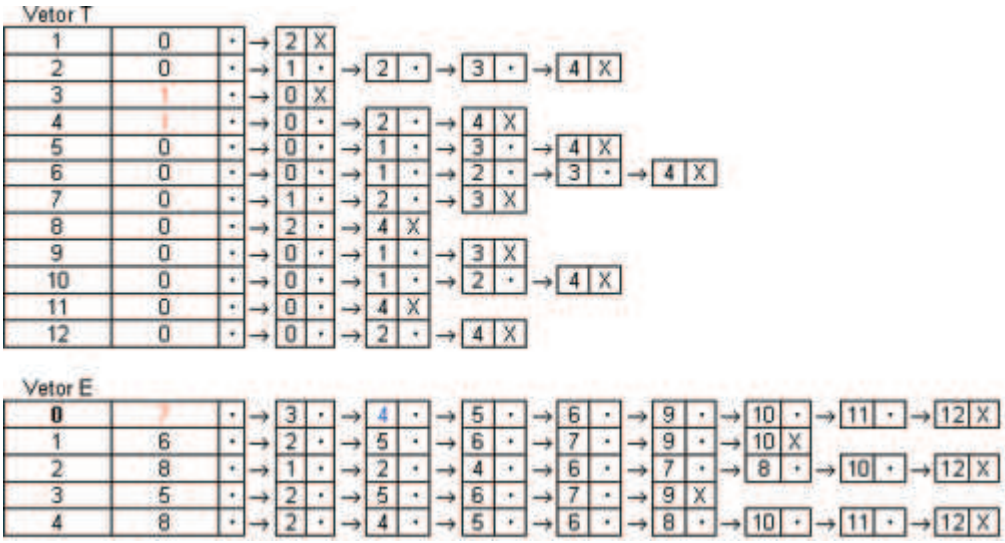

Figura 11: Marca-se o *flag* do elemento 4 no vetor T.

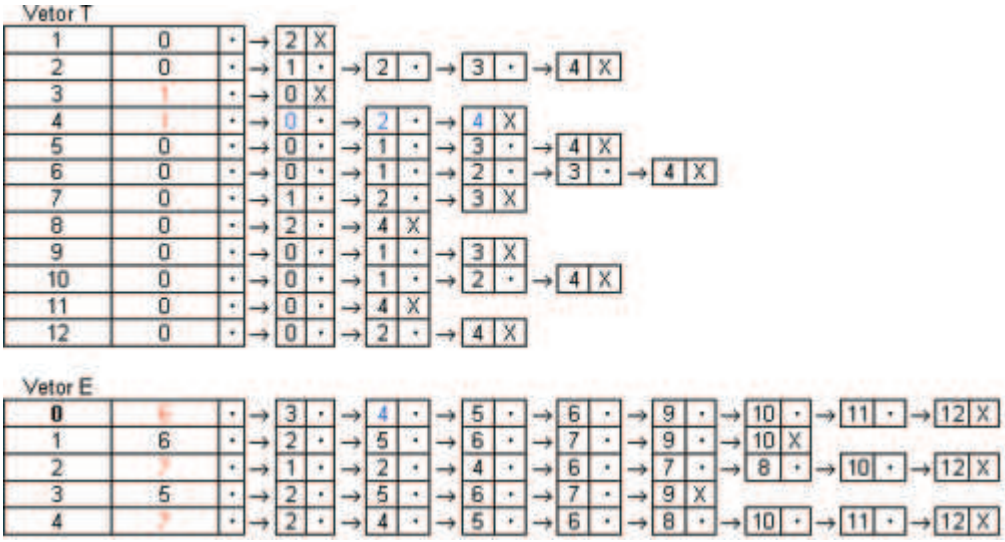

Figura 12: Percorre-se a lista ligada do elemento 4 no vetor T. Decrementando os respectivos contadores do vetor E.

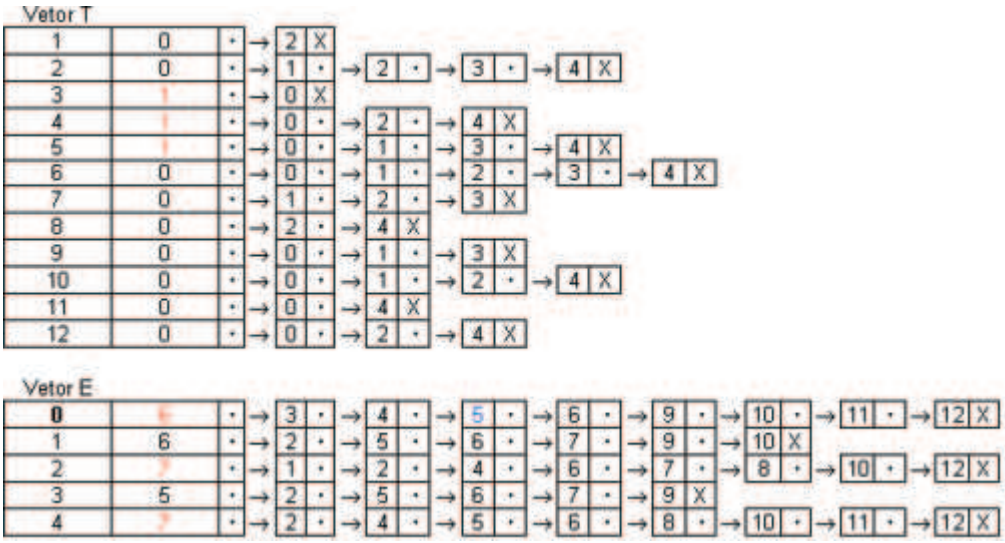

Figura 13: Marca-se o *flag* do elemento 5 no vetor T.

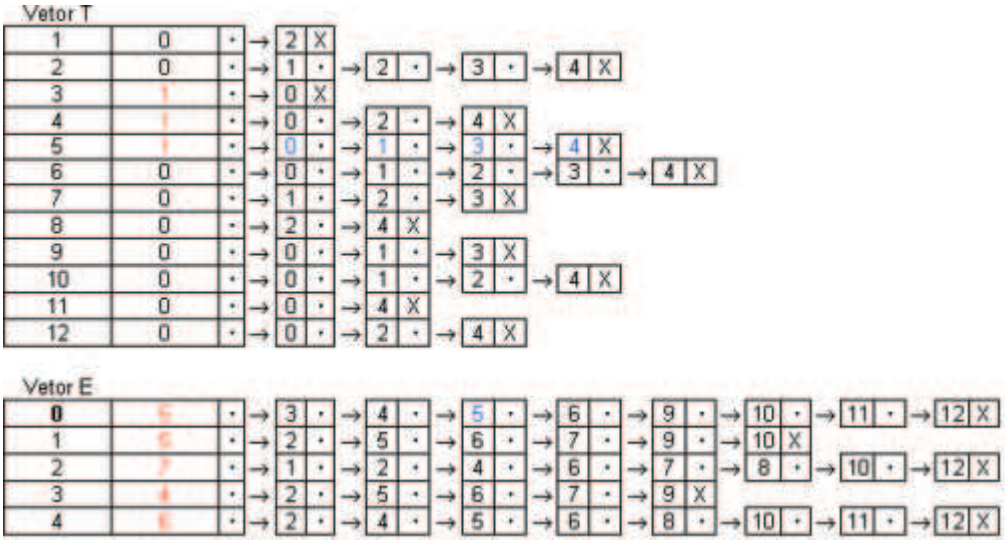

Figura 14: Percorre-se a lista ligada do elemento 5 no vetor T. Decrementando os respectivos contadores do vetor E.

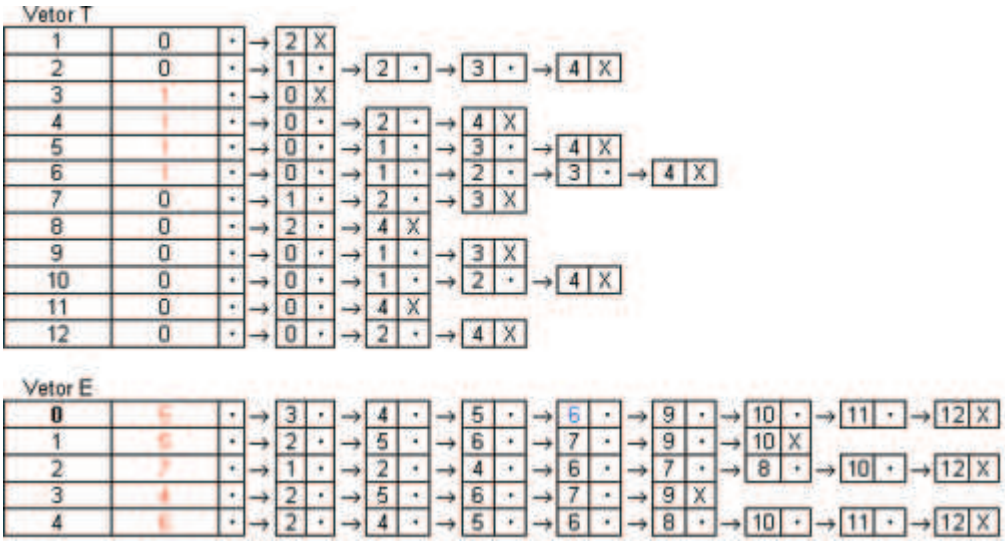

Figura 15: Marca-se o *flag* do elemento 6 no vetor T.

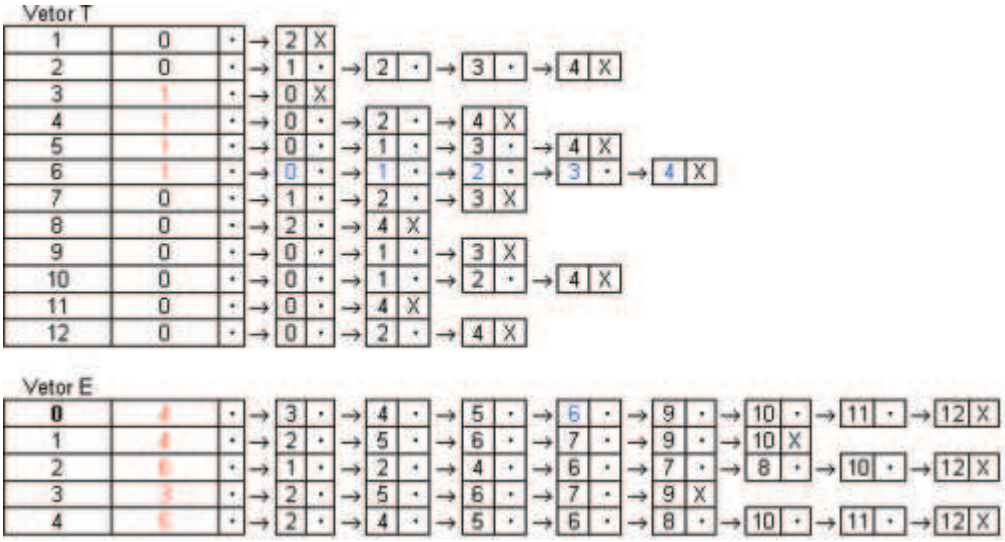

Figura 16: Percorre-se a lista ligada do elemento 6 no vetor T. Decrementando os respectivos contadores do vetor E.

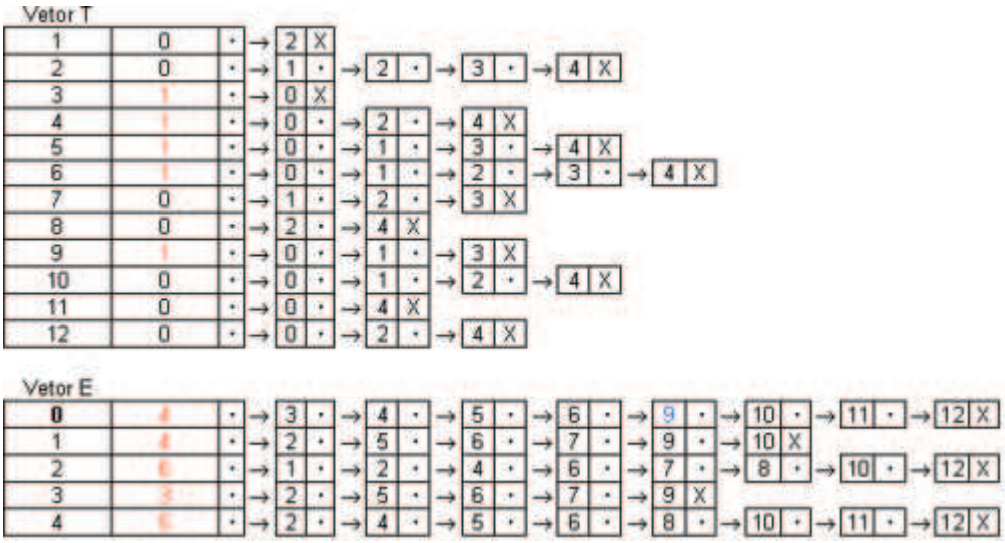

Figura 17: Marca-se o *flag* do elemento 9 no vetor T.

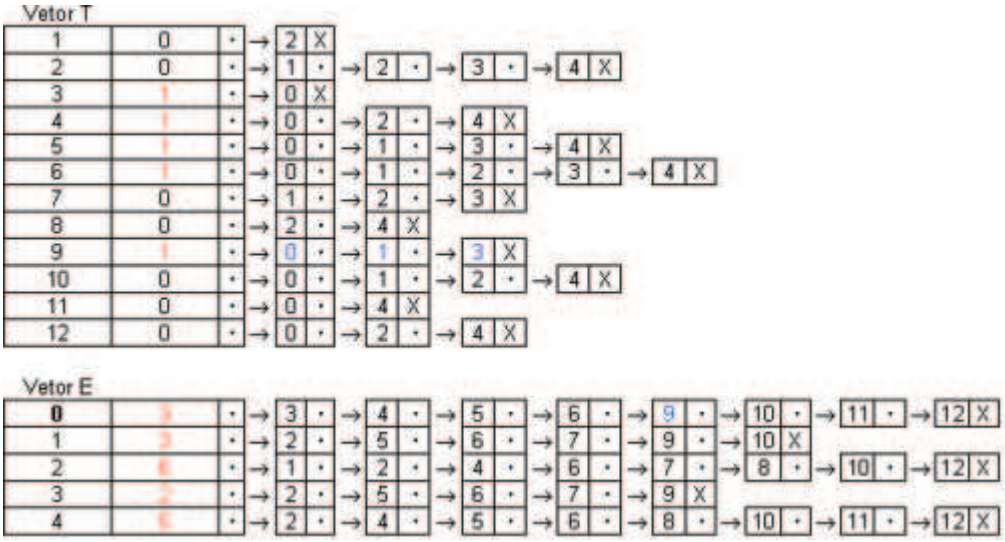

Figura 18: Percorre-se a lista ligada do elemento 9 no vetor T. Decrementando os respectivos contadores do vetor E.

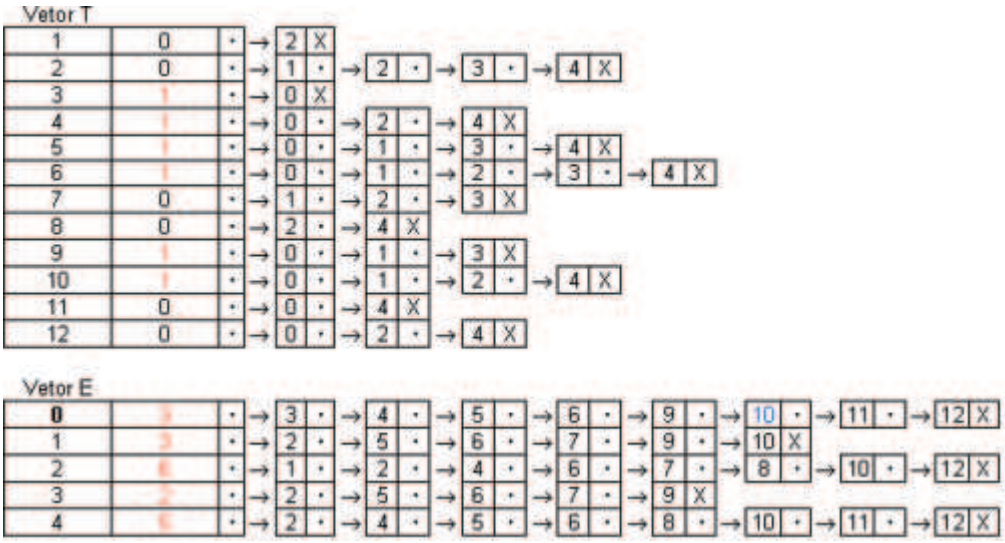

Figura 19: Marca-se o *flag* do elemento 10 no vetor T.

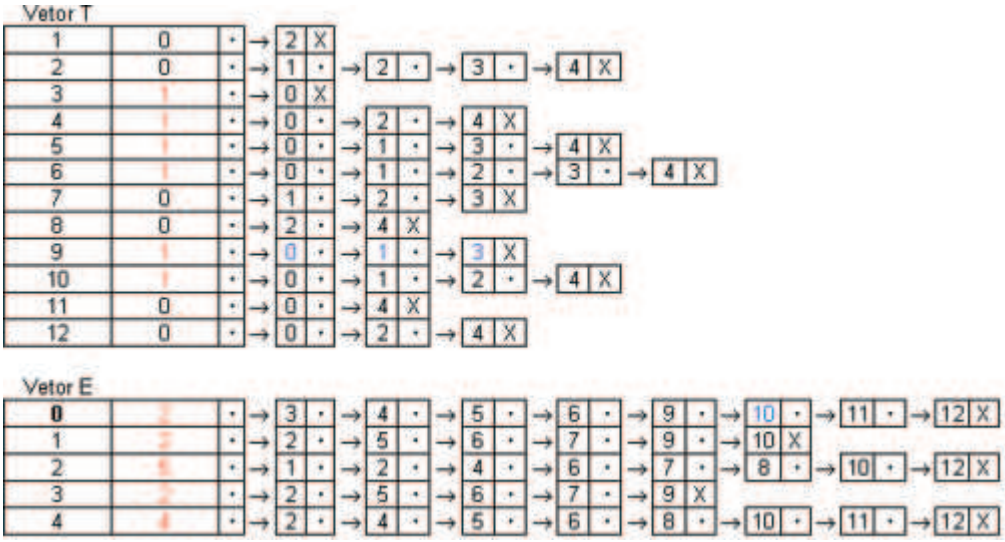

Figura 20: Percorre-se a lista ligada do elemento 10 no vetor T. Decrementando os respectivos contadores do vetor E.

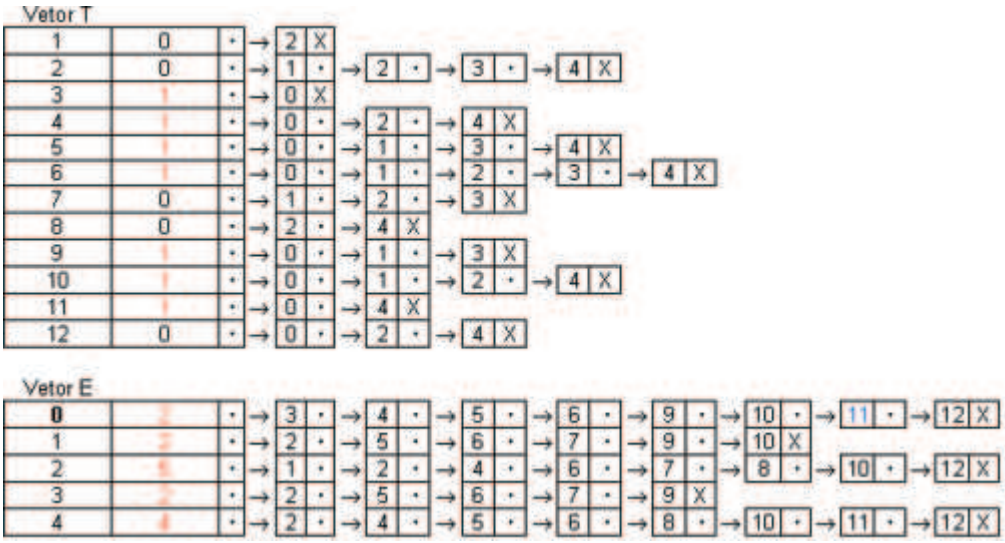

Figura 21: Marca-se o *flag* do elemento 11 no vetor T.

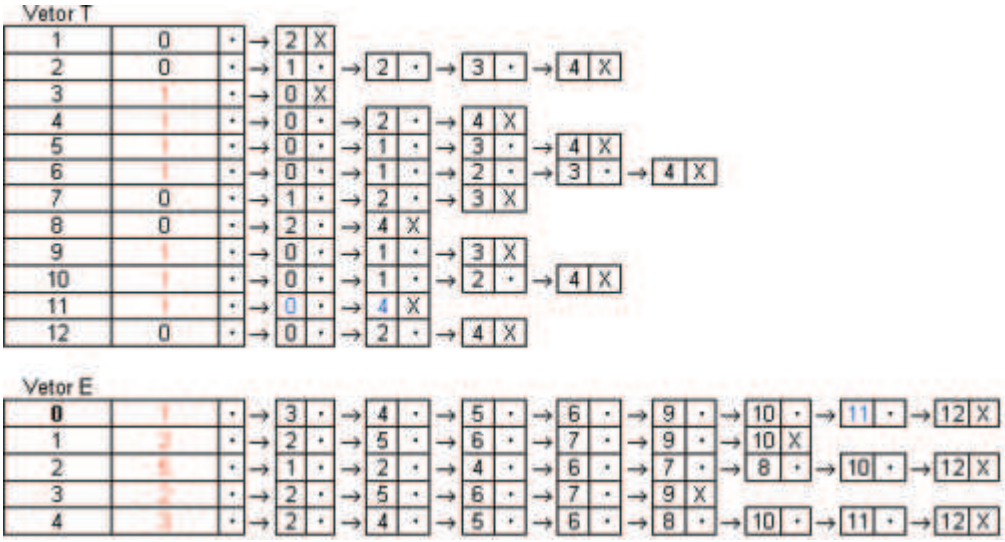

Figura 22: Percorre-se a lista ligada do elemento 11 no vetor T. Decrementando os respectivos contadores do vetor E.

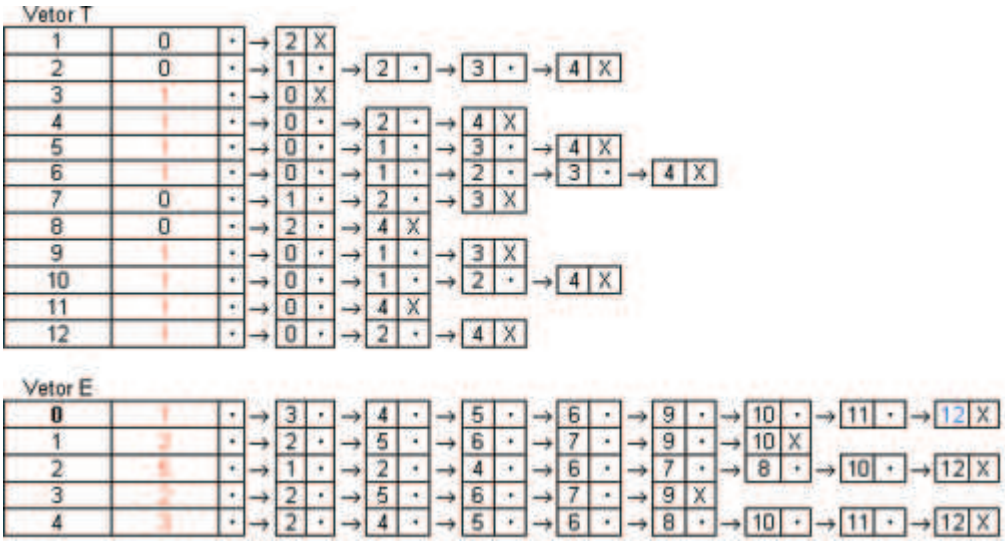

Figura 23: Marca-se o *flag* do elemento 12 no vetor T.

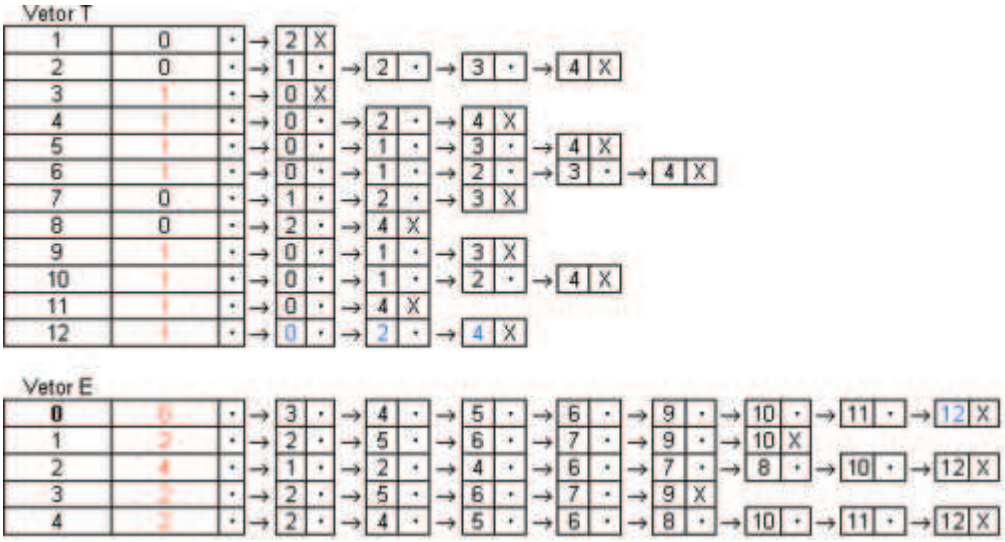

Figura 24: Percorre-se a lista ligada do elemento 12 no vetor T. Decrementando os respectivos contadores do vetor E.

Neste ponto o conjunto $E' = \{0\}$, como $\max\{\text{"contadores do vetor E"}\} \neq 0$ então deve-se continuar com o algoritmo.

O próximo valor a ser escolhido é o elemento 2 do vetor E, pois número de ocorrências do elemento 2 nos subconjuntos restantes é 4, sendo este a maior frequência.

Desta forma, a figura 25 apresenta o início da iteração para o elemento 2:

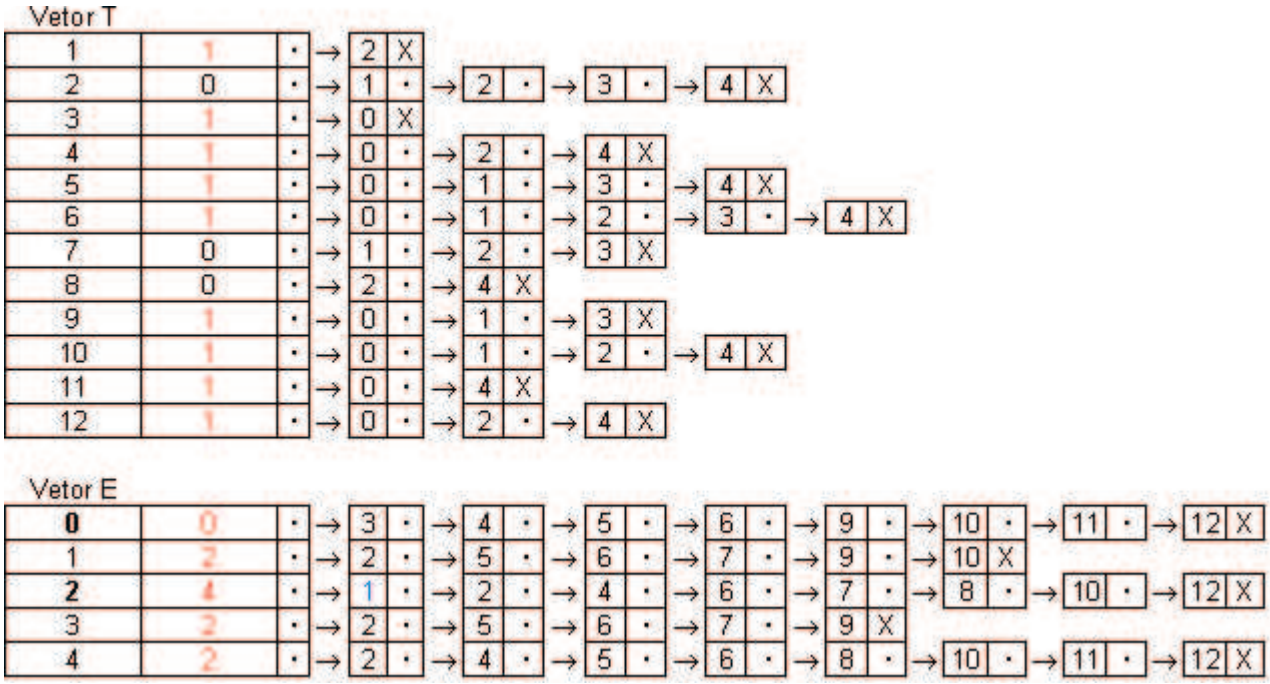

Figura 25: O elemento escolhido é 2 com 4 ocorrências.

Aplicando-se o algoritmo para esta lista ligada até o final do mesmo, isto é, para os elementos:

- 1
- 2
- 4
- 6
- 7
- 8
- 10
- 12

a estrutura de dados final terá o seguinte aspecto representado pela figura 27:

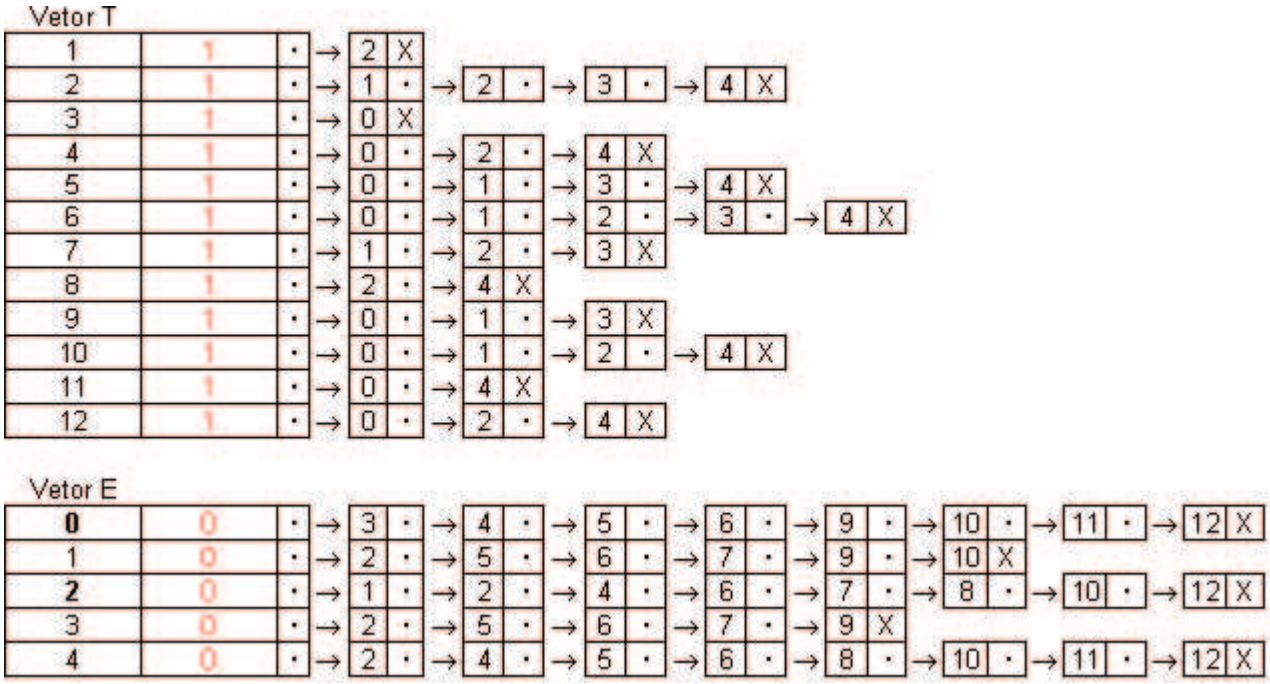

Figura 26: Situação final da estrutura de dados.

Como max{"contadores do vetor E"} $= 0$ [5] então o algoritmo aproximado pára neste ponto e o conjunto $E' = \{0, 2\}$ é a transversal mínima, cuja razão de aproximação é $\ln|\mathcal{S}|+1$ [10] (conforme visto na seção "Um algoritmo seqüêncial").

6 Uma interpretação do resultado numérico

Uma possível interpretação do resultado numérico obtido será apresentado a seguir.

Voltando a figura inicial do problema:

[5] Esta situação significa que todos os subconjuntos já foram "cobertos".

	Verdura Fresca	Comida Congelada	Frutas	Salgadinhos	Refrigerantes	**Legumes Frescos**
	0	1	2	3	4	
1	0	0	0	0	0	**0**
2	0	0	1	0	0	**1**
3	0	1	0	1	0	**0**
4	0	1	1	1	1	**1**
5	1	0	0	0	0	**1**
6	1	0	1	0	1	**1**
7	1	1	0	1	1	**1**
8	1	1	1	1	1	**1**

Figura 27: Dados de perfis de consumo interpretados inicialmente.

Observando os dados iniciais, e analisando o resultado do algoritmo aproximado para a transversal mínima $\{0, 2\}$, pode-se reduzir a tabela conforme a figura 28:

	Verdura Fresca	Frutas	**Legumes Frescos**
	0	2	
1	0	0	**0**
2	0	1	**1**
5	1	0	**1**
6	1	1	**1**

Figura 28: Trazendo apenas as colunas referentes aos elementos 0 e 2.

Nomeando a coluna 0 como x_0, a coluna 2 como x_2 e a coluna "Legumes Frescos"como x, pode-se construir a seguinte expressão pela tabela verdade:

$$x = \overline{x_0}.x_2 + x_0.\overline{x_2} + x_0x_2$$

$$x = x_0 + x_2$$

Ou seja, x depende de x_0 e x_2.

Este resultado poderia ser interpretado como:

> "Se um cliente comprar 'Verduras Frescas' ou comprar 'Frutas' então ele comprará 'Legumes Frescos'."

7 Uma paralelização do algoritmo seqüêncial

Uma paralelização a partir do algoritmo seqüêncial, baseado no modelo CGM (*Coarse-Grained Multicomputer* ilustrado anteriormente será ilustrado a seguir [8].

A partir dos subconjuntos obtidos inicialmente:

$$
\begin{aligned}
T_1 &= \{2\} \\
T_2 &= \{1,2,3,4\} \\
T_3 &= \{0\} \\
T_4 &= \{0,2,4\} \\
T_5 &= \{0,1,3,4\} \\
T_6 &= \{0,1,2,3,4\} \\
T_7 &= \{1,2,3\} \\
T_8 &= \{2,4\} \\
T_9 &= \{0,1,3\} \\
T_{10} &= \{0,1,2,4\} \\
T_{11} &= \{0,4\} \\
T_{12} &= \{0,2,4\}
\end{aligned}
$$

É necessário, dividir estes subconjuntos entre os processadores. Devido ao número de variáveis x_i ser pequena, neste caso é 5, será usado apenas 2 processadores.

Chamando de P_1 o primeiro processador e P_2 o segundo processador, tem-se a seguinte divisão dos subconjuntos, baseados no conteúdo dos elementos, isto é, P_1 receberá os elementos 0 e 1; P_2 receberá os elementos 3, 4 e 5, conforme divisão ilustrada abaixo:

Processador P_1 recebe:

$$
\begin{aligned}
T_1 &= \{\} \\
T_2 &= \{1\} \\
T_3 &= \{0\} \\
T_4 &= \{0\} \\
T_5 &= \{0,1\} \\
T_6 &= \{0,1\} \\
T_7 &= \{1\}
\end{aligned}
$$

$$T_8 = \{\}$$
$$T_9 = \{0, 1\}$$
$$T_{10} = \{0, 1\}$$
$$T_{11} = \{0\}$$
$$T_{12} = \{0\}$$

Processador P_2 recebe:

$$T_1 = \{2\}$$
$$T_2 = \{2, 3, 4\}$$
$$T_3 = \{\}$$
$$T_4 = \{2, 4\}$$
$$T_5 = \{3, 4\}$$
$$T_6 = \{2, 3, 4\}$$
$$T_7 = \{2, 3\}$$
$$T_8 = \{2, 4\}$$
$$T_9 = \{3\}$$
$$T_{10} = \{2, 4\}$$
$$T_{11} = \{4\}$$
$$T_{12} = \{2, 4\}$$

Feita a divisão dos subconjuntos, cada processador recebe seu conjunto de subconjuntos e deve construir e inicializar a estrutura de dados, conforme a figura 29:

Vetor T

1	0	X
2	0	X
3	0	X
4	0	X
5	0	X
6	0	X
7	0	X
8	0	X
9	0	X
10	0	X
11	0	X
12	0	X

Vetor E

0	0	X
1	0	X

Vetor T

1	0	X
2	0	X
3	0	X
4	0	X
5	0	X
6	0	X
7	0	X
8	0	X
9	0	X
10	0	X
11	0	X
12	0	X

Vetor E

2	0	X
3	0	X
4	0	X

Figura 29: Situação inicial da estrutura de dados em cada processador: P_1 e P_2, respectivamente.

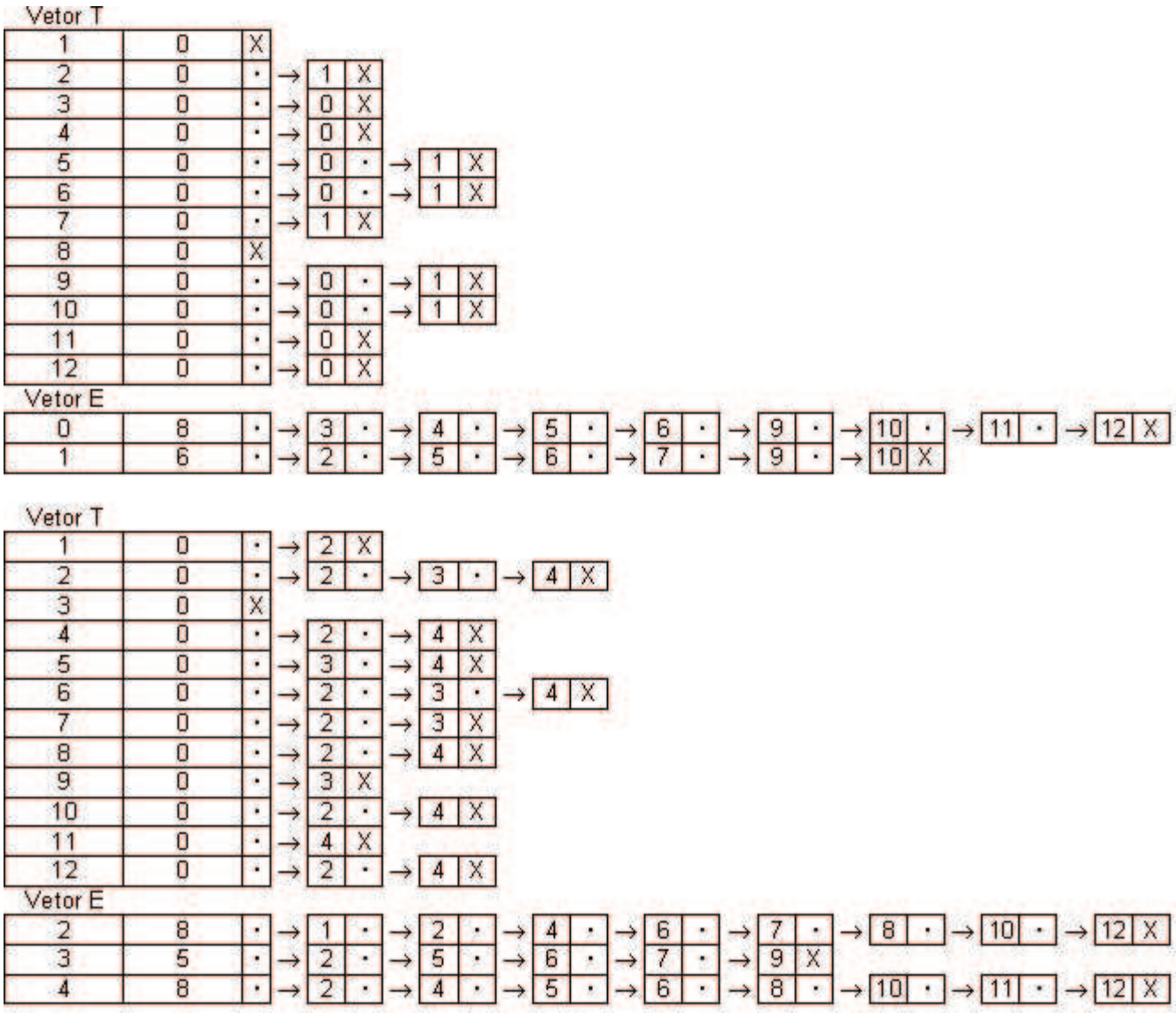

Figura 30: Situação após a carga inicial da estrutura de dados em cada processador: P_1 e P_2, respectivamente.

Após a carga inicial dos dados nas estruturas, é necessário eleger um processador, que terá como uma das funções decidir (processador "decisor") se o algoritmo deve parar, através da seguinte lógica:

1. O processador "decisor"faz $E' = \varnothing$ (este conjunto conterá a transversal mínima).

2. O processador "decisor"recebe de cada um dos processadores o max{"contadores do vetor E"} para cada estrutura de dados.

3. Enquanto o processador "decisor"recebe algum max{"contadores do vetor E"} $\neq 0$ ele faz: [6]

   - Verifica qual o maior valor recebido dos processadores.
   - Para o processador escolhido, solicita a lista ligada e o elemento associado e correspondente.
   - O processador "decisor", de posse desta informação, comunica aos demais processadores esta lista ligada (faz *broadcasting* da lista) e faz: $E' = E' \cup \{e\}$.
   - Cada processador ao receber esta informação deve recalcular a sua estrutura de dados como no algoritmo seqüêncial guloso, com a diferença de que a lista ligada a ser trabalhada é a lista ligada recebida do processador "decisor".
   - Ao final da fase de computação cada processador deve enviar novamente ao processador "decisor"o novo max{"contadores do vetor E"}.

A seguir ilustra-se a aplicação da lógica descrita anteriormente:

- Seja o processador "decisor", o processador 1 (P_1) e $E' = \varnothing$.
- Partindo das estruturas já carregadas em cada um dos processadores (Figura 30), cada processador envia para o processador "decisor"o seu max{"contadores do vetor E"}:

  P_1 enviará o valor 8 (referente ao contador do elemento 0 do vetor E na estrutura de dados do processador P_1) ao processador "decisor".

  P_2 enviará o valor 8 (referente ao contador do elemento 2 do vetor E na estrutura de dados do processador P_2) ao processador "decisor".

- O processador "decisor"ao receber estas informações, solicita ao processador P_1 a lista ligada e o elemento associado (neste caso o elemento $e = 0$) repassa a informação ao processador P_2 (faz *broadcasting* da informação) e faz $E' = E' \cup \{0\}$.
- De posse desta informação, os processadores recalcularão suas estruturas de dados como na figura 31, neste caso a lista ligada é formada pelos elementos: 3, 4, 5, 6, 9, 10, 11 e 12.

[6] Como o processador "decisor"também tem sua própria estrutura de dados então também recebe de si próprio o seu "max".

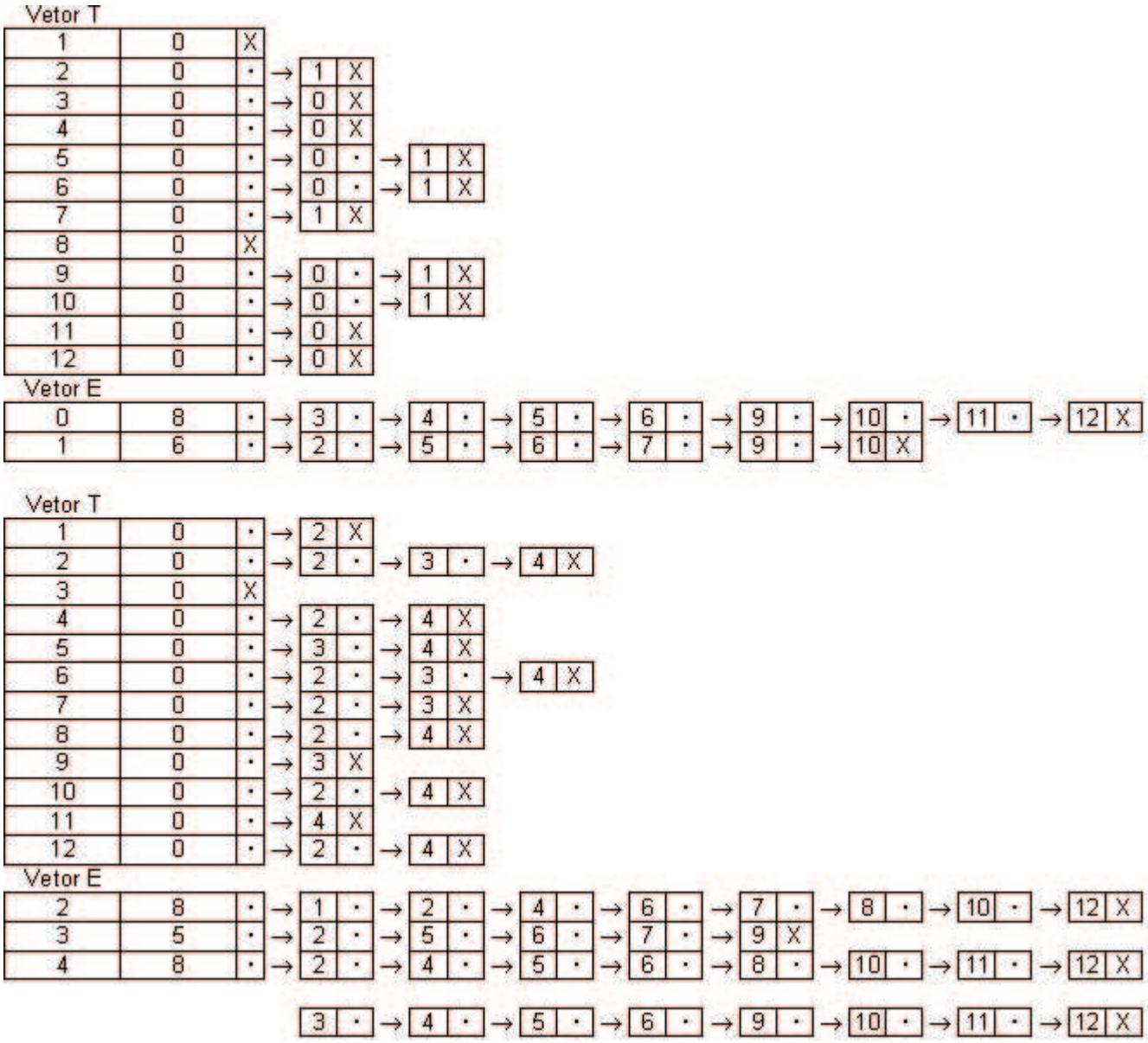

Figura 31: P_1 irá trabalhar com a lista ligada do elemento 0 (do vetor E) e P_2 irá trabalhar com a lista ligada recebida.

- Após recalcular as estruturas de dados pelo algoritmo seqüêncial guloso, as estruturas de dados ficarão como ilustra a figura 32:

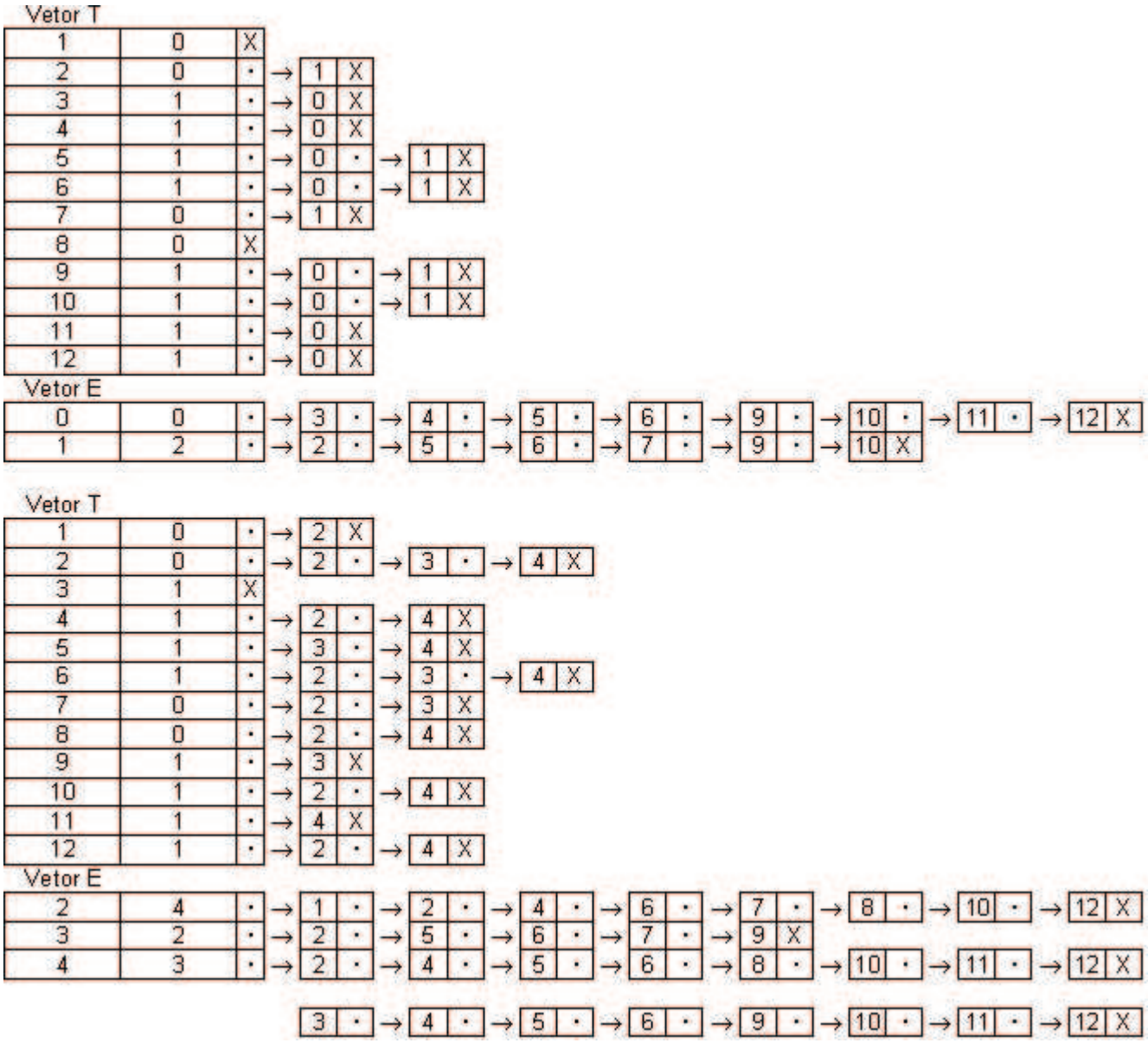

Figura 32: Estruturas de dados de P_1 e P_2 respectivamente após a atualização das estruturas.

- Com a etapa de computação concluída, cada processador envia para o processador "decisor"o seu novo max{"contadores do vetor E"}:

  P_1 enviará o valor 2 (referente ao contador do elemento 1 do vetor E na estrutura de dados do processador P_1) ao processador "decisor".

  P_2 enviará o valor 4 (referente ao contador do elemento 2 do vetor E na estrutura de dados do processador P_2) ao processador "decisor".

- O processador "decisor"ao receber estas informações, solicita ao processador P_2 a lista ligada e o elemento associado (neste caso o elemento $e = 2$) repassa a informação ao processador P_1 (faz *broadcasting* da informação) e faz $E' = E' \cup \{2\}$ (resultando em $E' = \{0, 2\}$).

- De posse desta informação, os processadores recalcularão suas estruturas de dados como na figura 33, neste caso a lista ligada é formada pelos elementos: 1, 2, 4, 6, 7, 8, 10 e 12.

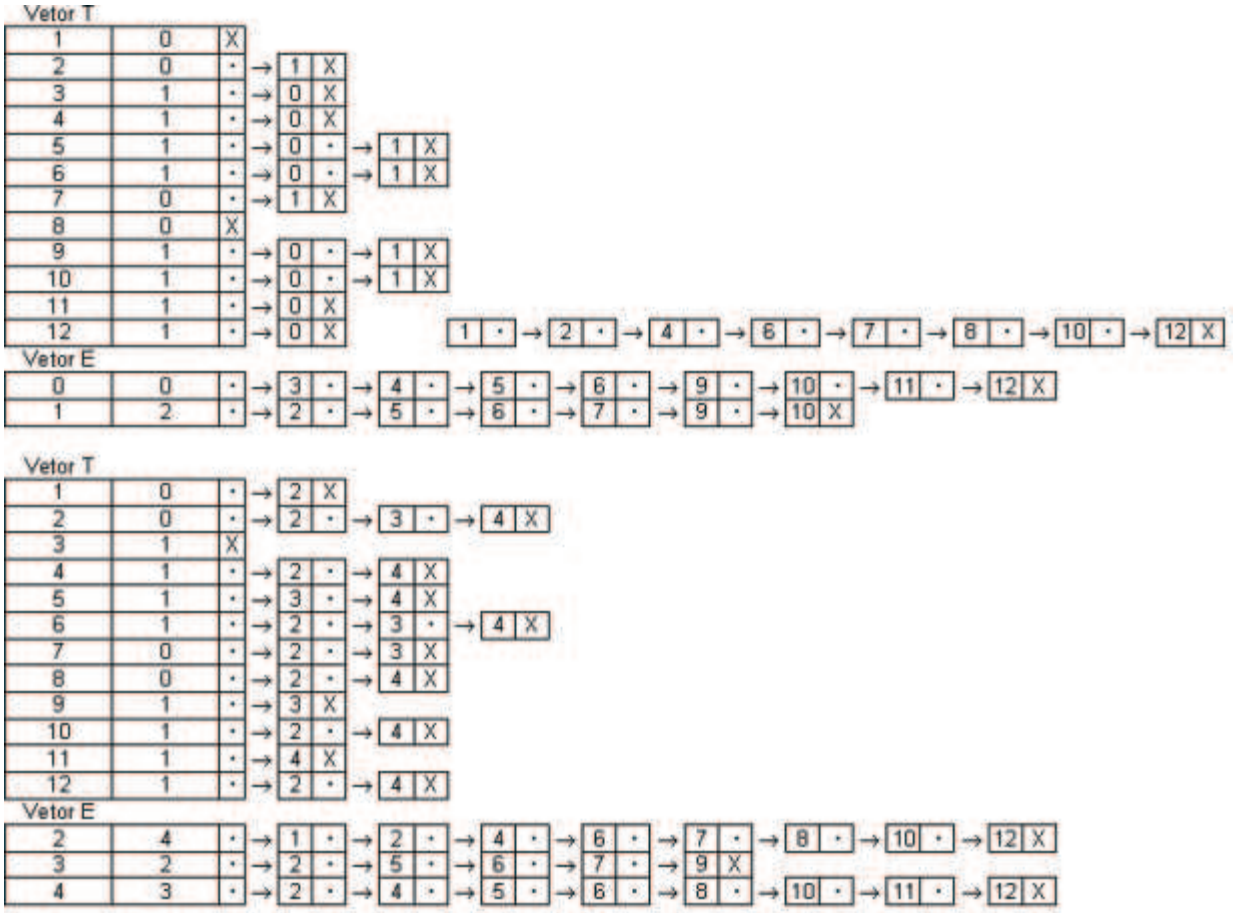

Figura 33: P_1 irá trabalhar com a lista ligada recebida e P_2 irá trabalhar com a lista ligada do elemento 2 (do vetor E).

- Após recalcular as estruturas de dados pelo algoritmo seqüêncial guloso, as estruturas de dados ficarão como ilustra a figura 34:

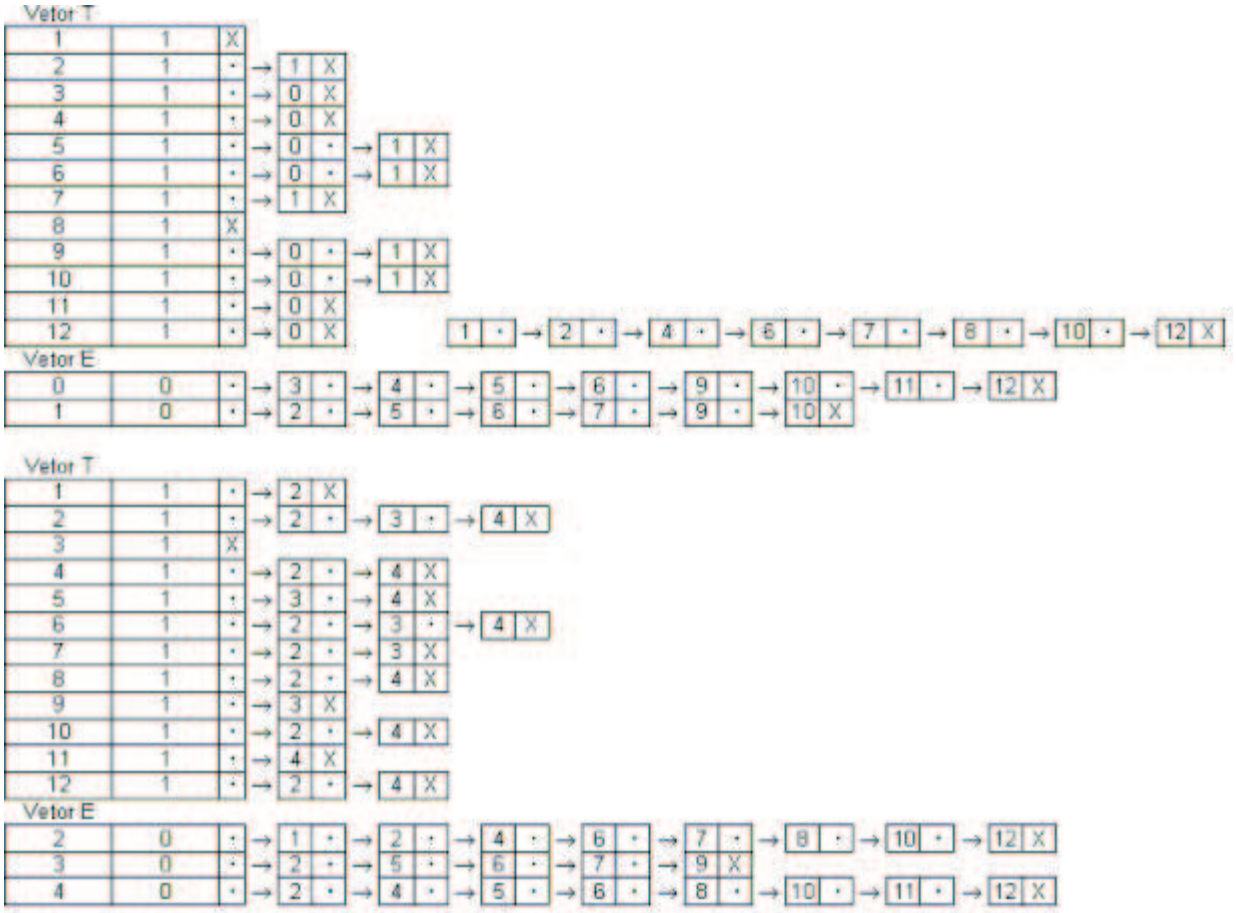

Figura 34: Estruturas de dados de P_1 e P_2 respectivamente após a atualização das estruturas.

- Após o término desta etapa de computação cada processador envia para o processador "decisor"o seu novo max{"contadores do vetor E"}:

  P_1 enviará o valor 0 (observe que todos os contadores do vetor E na estrutura de dados do processador P_1 são nulos) ao processador "decisor".

  P_2 enviará o valor 0 (observe que todos os contadores do vetor E na estrutura de dados do processador P_1 são nulos) ao processador "decisor".

- Como o processador "decisor"recebe apenas valores nulos, este decide que o algoritmo deve terminar e portanto $E' = \{0, 2\}$ é a transversal mínima.

É interessante observar que o número de rodadas de comunicação é a cardinalidade da transversal mínima encontrada, ou seja, o número máximo de rodadas de

comunicação é o tamanho máxido do vetor E, que compõem os possíveis elementos da trasnversal mínima.

Sejam:

- m o número de subconjuntos T de E (no exemplo anterior $m = 12$).
- n o número de elementos do conjunto E (no exemplo anterior $n = 5$).
- p o número de processadores.
- E' a transversal mínima encontrada. $|E'| = \varphi$

Na execução do algoritmo seqüêncial é possível provar que o tempo de execução é $O(mn^2)$.

Na execução do algoritmo paralelo é possível provar que o tempo de execução é $O(\frac{m.n^2}{p})$ com apenas $O(\varphi)$ rodadas de comunicação.

8 Máquinas paralelas para implementação

Foram estudados as características de algumas máquinas paralelas no IME, para possível implementação dos algoritmos para o problema da transversal mínima.

8.1 Máquina da classe *cluster*

A classe de máquinas paralelas do tipo *Beowulf* foi idealizado em 1994 por Thomas Sterling e Don Becker da CESDIS (*Center of Excellence in Space Data and Information Sciences*) - NASA. Foram utilizadas 16 máquinas Intel 486DX-4 juntamente com o sistema operacional Linux Debian. A principal idéia destas classes de máquinas era de compor um sistema de grande poder computacional baseado em máquinas de baixo custo encontrado facilmente no mercado.

O IME/USP possui uma máquina do tipo *cluster Beowulf* composto por 16 PC's (nós).

Cada nó foi construído com as seguintes características:

- Processador: 1.2Ghz AMD Thunderbird Athlon, 256Kbytes de memória cache L2.

- Memória RAM de 768 Mbytes PC133 SDRAM.
- Disco rígido: 73 Gbytes ATA100 7.200 rpm.
- NIC (*network interface card*): 3Com fast ethernet de 100Mbits/s.
- Sistema Operacional: Linux Debian kernel 2.2

A interligação de cada nó é feito via *switcher 3Com superstack 3300*, conforme diagrama abaixo:

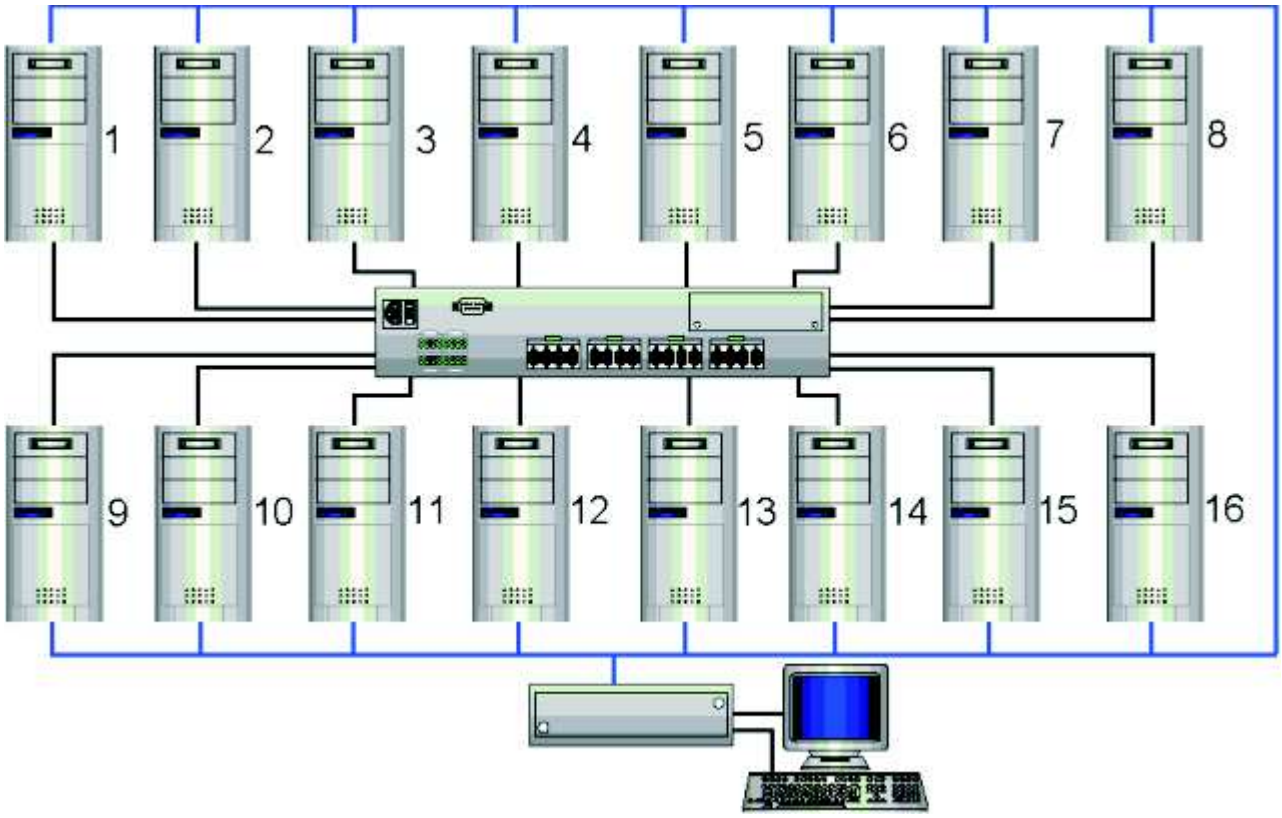

Figura 35: Máquina *Beowulf* do IME: o Biowulf.

O acesso a esta máquina pela rede IME pode ser feita através no primeiro nó do *cluster*, de nome tiramissu.ime.usp.br.

Atualmente esta máquina está alocada para o projeto CAGE (*Cooperation for Analysis of Gene Expression*) patrocinado pela FAPESP e gerenciada pela professora Martha Torres.

8.2 Novas máquinas multiprocessadas da Sun Microsystems

Está previsto para os próximos meses a chegada e instalação de novas máquinas multiprocessadas, patrociandas pela Sun Microsystems. Estas máquinas possuem alto poder computacional tendo como algumas características:

1. Sun Enterprise 3500
   - 4 processadores Superscalar Sparc V.9 600 Mhz
   - 8 Gbytes RAM
2. Sun Enterprise 4500
   - 10 processadores 250 Mhz
   - 10 Gbytes RAM
   - 1 Terabyte HD
3. Sun Fire V880
   - 32 processadores UltraSparc III 800Mhz
   - 32 Gbytes RAM

9 Conclusão

O estudo do problema da transversal mínima, sua solução por algoritmos de aproximação seqüêncial e paralelo, implementados em máquinas reais pode contribuir em inúmeras áreas como: computação gráfica, bioinformática e *datamining* entre outras.

Referências

[1] T. E. Ideker, V. Thorsson and R. M. Karp. *Discovery of Regulatory Interactions through Perturbation: Inference and Experimental Design*. Pacific Symposium on Biocomputing, 2000, pp. 1-12.

[2] C. G. Fernandes, F. K. Miyazawa e M. Cerioli (editores). *Uma Introdução Sucinta a Algoritmos de Aproximação*. 23.o Colóquio Brasileiro de Matemática. IMPA. 2001.

[3] M. J. Simmons e D. P. Snustad. *Principles of genetics*. Wiley, 2nd edition, 1999.

[4] J. Tsang. *Gene expression, DNA arrays, and genetic networks*.

[5] S. Fuhrman, S. Liang e R. Somogyi. *Reveal, a general reverse engineering algorithm for inference of genetic network architectures*. Pacific Symposium on Biocomputing 3:18-29, 1998.

[6] P. D. haeseleer, S. Liang e R. Somogyi. *Gene expression data analysis and modeling*. Tutorial notes from Pacific Symposium on Biocomputing, 1999.

[7] D.S. Hochbaum, editor. *Aproximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.

[8] D.P. Ruchkys e S.W. Song. *Processamento Paralelo para Análise da Expressão Gênica.*, 2002.

[9] S. Jha, O. Sheyner e J. M. Wing. *Minimization and reliability analyses of attack graphs.* IEEE Symposium on Security and Privacy, 2002.

[10] T.H. Cormen, C.E. Leiserson e R.L. Rivest. *Introduction to Algorithms*. MIT Press, 1999.