

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**

**DEPARTAMENTO DE INFORMÁTICA**

**MESTRADO EM INFORMÁTICA**

***Uma abordagem em Análise de Cluster para problemas de  
agrupamento de áreas florestais***

Aluno: Alexsandro Afonso Moura

Orientadores: Hannu Tapio Ahonen

Arlindo Gomes Alvarenga

Vitória, 27 de agosto de 2004

## ÍNDICE

|            |   |    |
|------------|---|----|
| Capítulo 1 | Introdução.....   | 1  |
| Capítulo 2 | Caracterização do Problema .....                          | 3  |
| Capítulo 3 | Ferramentas de Agrupamento.....                           | 6  |
| 3.1.       | Métodos Hierárquicos.....                                 | 7  |
| 3.2.       | Métodos de Partição .....                                 | 11 |
| 3.3.       | Métodos com abordagem em teoria dos grafos.....           | 13 |
| 3.3.1      | Algoritmo HCS ( <i>highly connected subgraph</i> ).....   | 16 |
| 3.3.1.1.   | Algoritmo de Corte Mínimo .....                           | 23 |
| 3.3.2      | Agrupamento através de <i>Minimum Spanning Tree</i> ..... | 28 |
| 3.3.2.1.   | Algoritmo MST ( <i>Minimum Spanning Tree</i> ) .....      | 28 |
| Capítulo 4 | Estratégia de Solução .....                               | 31 |
| 4.1.       | Implementação dos algoritmos.....                         | 33 |
| 4.2.       | Comparação dos métodos .....                              | 37 |
| 4.3.       | Novos critérios .....                                     | 41 |
| 4.3.1      | Novo exemplo.....   | 43 |
| 4.4.       | Melhorias importantes .....                               | 44 |
| 4.5.       | Modelo de classes.....                                    | 45 |
| 4.5.1      | Classes para o HCS.....                                   | 45 |
| 4.5.2      | Classes para o MST- <i>Clustering</i> invertido .....     | 49 |
| Capítulo 5 | Testes Numéricos.....                                     | 51 |
| Capítulo 6 | Conclusões.....   | 62 |
| Capítulo 7 | Apêndice A – Exemplos de conjuntos de testes.....         | 66 |

## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 - Mapa de áreas florestais .....                               | 3  |
| Figura 2- Diagrama em árvore.....                                       | 7  |
| Figura 3 -Diagrama por <i>single linkage</i> .....                      | 10 |
| Figura 4 - Diagrama com segunda escolha .....                           | 11 |
| Figura 5 – Algoritmo K-Means .....                                      | 12 |
| Figura 6 – Algoritmo de Herwig et ali.....                              | 13 |
| Figura 7 – Algoritmo CAST.....  | 15 |
| Figura 8 – Algoritmo <i>FormKernels</i> .....                           | 17 |
| Figura 9 – Algoritmo CLICK .....  | 18 |
| Figura 10 – Algoritmo HCS .....   | 19 |
| Figura 11 – Exemplo de execução do HCS.....                             | 20 |
| Figura 12 – Algoritmo HCS incrementado .....                            | 22 |
| Figura 13 – Exemplo de entrada para o problema de corte mínimo.....     | 24 |
| Figura 14 – “Identificação” de nós .....                                | 25 |
| Figura 15 - Algoritmo de Corte Mínimo por Identificação de Nós.....     | 25 |
| Figura 16 – Início do primeiro passo do <i>minimum cut</i> .....        | 26 |
| Figura 17 – Final do primeiro passo do <i>minimum cut</i> .....         | 26 |
| Figura 18 – Segundo passo do algoritmo de minimum-cut .....             | 27 |
| Figura 19 – Último passo (n-1) do algoritmo de <i>minimum cut</i> ..... | 27 |
| Figura 20 – Algoritmo de Kruskal.....                                   | 29 |
| Figura 21 – Algoritmo <i>Partitional MST</i> .....                      | 30 |
| Figura 22 – Algoritmo de Agrupamento via HCS.....                       | 34 |
| Figura 23 – Algoritmo <i>GeraGrafoSimilaridade</i> .....                | 35 |
| Figura 24 – Algoritmo <i>MST-Clustering</i> Invertido .....             | 36 |
| Figura 25 – Conjunto de plantios 1.....                                 | 37 |

|  |    |
|--|----|
| Figura 26 – Matriz de Adjacências 1 .....  | 37 |
| Figura 27 – Grafo de Similaridade 1.....   | 38 |
| Figura 28 – <i>Clusters</i> por HCS 1 .....  | 38 |
| Figura 29 – Máxima <i>Spanning Tree</i> 1 .....  | 38 |
| Figura 30 – Conjunto de plantios 2.....  | 39 |
| Figura 31 – Matriz de Adjacências 2 .....  | 39 |
| Figura 32 – Grafo de Similaridade 2.....   | 40 |
| Figura 33 – <i>Clusters</i> por HCS 2 .....  | 40 |
| Figura 34 – Máxima <i>SpanningTree</i> 2 .....   | 40 |
| Figura 35 – Conjunto de Plantios 3.....  | 43 |
| Figura 36 – Matriz de Adjacências 3 .....  | 43 |
| Figura 37 – Grafo de Similaridade 3.....   | 44 |
| Figura 38 – Máxima <i>Spanning Tree</i> 3 .....  | 44 |
| Figura 39 – Modelo de Classes para HCS (Elemento e Vértice) .....  | 46 |
| Figura 40 – Modelo de Classes para HCS.....  | 47 |
| Figura 41 – Modelo de Classes para MST- <i>Clustering</i> invertido.....   | 50 |
| Figura 42 – Exemplo de conjunto de 50 plantios .....   | 52 |
| Figura 43 – Gráfico de tempo de execução para conjuntos de 100 plantios .....                                    | 59 |
| Figura 44 - Gráfico de tempo de execução para conjuntos de 50 plantios.....                                      | 59 |
| Figura 45 - Gráfico do número de <i>clusters</i> e <i>singletons</i> gerados para conjuntos de 100 plantios .... | 60 |
| Figura 46 - Gráfico do número de <i>clusters</i> e <i>singletons</i> gerados para conjuntos de 50 plantios ..... | 60 |
| Figura 47 - Gráfico de similaridade dos <i>clusters</i> gerados para conjuntos de 100 plantios.....              | 61 |
| Figura 48 - Gráfico de similaridade dos <i>clusters</i> gerados para conjuntos de 50 plantios.....               | 61 |

## LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 1 – Similaridade de idade.....                               | 32 |
| Tabela 2 – Similaridade da classe de densidade.....                 | 32 |
| Tabela 3 – Similaridade do tipo de regeneração.....                 | 33 |
| Tabela 4 – Similaridade da localização.....                         | 33 |
| Tabela 5 – Ajuste de Produtividade na Idade.....                    | 41 |
| Tabela 6 – Similaridade de idade.....                               | 42 |
| Tabela 7 – Exemplo de matriz de adjacências de 50 plantios.....     | 53 |
| Tabela 8 - Dados dos testes nos conjuntos de 100 plantios.....      | 55 |
| Tabela 9 - Dados dos testes nos conjuntos de 50 plantios.....       | 57 |
| Tabela 16 – Matriz de adjacências do conjunto 1 de 50 plantios..... | 73 |
| Tabela 17 – Matriz de adjacências do conjunto 2 de 50 plantios..... | 74 |
| Tabela 18 – Matriz de adjacências do conjunto 3 de 50 plantios..... | 75 |

# Capítulo 1 Introdução

Uma empresa que produz materiais em grande quantidade e com características diferentes tem a necessidade de uma organização desses materiais ou matérias-primas, de modo que a empresa possa entregá-los ou usá-los com boa qualidade e em tempo.

Para o bom planejamento, todo o conjunto de produtos deve estar mapeado fornecendo informações necessárias para as tomadas de decisão. Quanto maior o detalhe das informações sobre os produtos, mais detalhada será a análise.

Sabendo-se onde estão sendo produzidos ou armazenados, como ordenar a produção ou o recebimento de suprimentos de determinados produtos no tempo mais propício para sua venda ou utilização? Este problema pode ser caracterizado com um problema de agrupamento ou *clustering*, onde serão criados grupos de produtos homogêneos para que o analista possa fazer um planejamento melhor.

Um exemplo é o controle das áreas florestais de uma empresa de fabricação de celulose. Normalmente, uma empresa desse ramo mantém uma área florestal enorme distribuída em áreas menores que podem estar bem distantes umas das outras. As árvores são plantadas em diferentes épocas e dependendo da localização podem ter um crescimento mais lento ou acelerado. Além disso, estudos genéticos e de fertilização vão gerando novas plantas cada vez melhores para a produção de celulose. Nessa floresta, a empresa deve colher uma quantidade de madeira de forma contínua e programada para suprir sua fábrica de celulose. Também deve se preocupar em maximizar o aproveitamento da madeira colhida, observando o período ideal para o corte, reduzir os gastos com maquinário e operação de corte procurando cortar madeira de áreas próximas em períodos subseqüentes e equilibrar a composição da madeira como material de fabricação para garantir a qualidade da celulose.

Dessa forma é necessário um bom mapeamento das características de cada grupo de árvores plantadas, pois em cima dessas características é que serão feitas as análises para definição dos grupos de corte ou colheita. Esse problema será visto mais detalhadamente no capítulo 2.

Existem vários estudos sobre *clustering*, principalmente nas áreas de estatística e aplicadas à medicina. Vamos estudar essas técnicas para definir uma boa abordagem para o problema.

O objetivo será fornecer uma solução para o problema especificado para auxiliar um planejamento de corte de madeira baseado na existência de grupos homogêneos dentro da área total plantada. Os grupos homogêneos ou *clusters* são resultado da aplicação da técnica de *clustering* ou agrupamento. Primeiramente, no Capítulo 3 , serão apresentadas algumas técnicas de *clustering* com suas abordagens, algoritmos ou heurísticas e alguns resultados ou exemplos de uso. Num segundo momento tomaremos a abordagem conhecida por HCS (*Highly Connected Subgraphs*) e seguiremos com sua implementação. Para efetivar essa parte, necessitaremos de alguns métodos auxiliares para mapear computacionalmente e manipular as informações dos elementos a serem agrupados, e ainda pode ser necessária alguma adaptação dos algoritmos usados.

Em seguida, direcionaremos a implementação para suportar dados similares a uma floresta real de madeira de corte - um estudo de caso - com a definição de uma estratégia para a formação dos grupos (características avaliadas, semelhanças e diferenças).

Finalizaremos apresentando, no Capítulo 5 , os resultados da aplicabilidade da solução para o problema comparando com uma abordagem que simule a análise humana para o mesmo caso e os resultados possíveis.

## Capítulo 2 Caracterização do Problema

No cenário de uma empresa que planta e colhe madeira para processamento figura a necessidade de uma organização das áreas florestais visando a época do corte, de modo que a fábrica receba produtos de boa qualidade e em tempo.

Observe na Figura 1, que a extensão da área plantada pode ser muito grande. A diversidade de tipos de madeira também.



**Figura 1 - Mapa de áreas florestais**

Para o bom planejamento toda a área florestal deve estar identificada, fornecendo as informações necessárias para as tomadas de decisão. Como diferentes regiões são plantadas e colhidas em tempos diferentes, as menores áreas homogêneas são chamadas de plantios. Os dados vão da localização e área de um plantio até a espécie, densidade da madeira e fator de crescimento, entre outros. Vejamos alguns exemplos:



- Idade – as árvores têm um tempo médio propício para a colheita, no caso do eucalipto é de seis a sete anos. Portanto, o primeiro passo para saber se um plantio pode ser cortado é avaliar sua idade.
- Localização do plantio – diferentes áreas podem estar prontas para o corte no mesmo período. Entretanto, a equipe de colheita não pode cobrir toda a área ao mesmo tempo e não é bom que faça um vai e vem dispendioso. Numa análise rápida do mapa, percebe-se a existência de quatro grandes aglomerados de área plantada. Uma pessoa pode já descartar a união de áreas de colheita que estejam em aglomerados diferentes, ou seja, a colheita deve ser feita por uma equipe diferente. Em um desses aglomerados é mais complicado tomar a decisão de quais áreas agrupar. Neste caso, outras condições devem ser consideradas.
- Proximidade entre plantios – plantios próximos podem ser cortados por uma mesma equipe em seqüência, o que torna esse um fator essencial, mas não simples. Apesar da proximidade geográfica, algum obstáculo, natural ou não, pode impedir o acesso rápido de uma área à outra. Por exemplo, pode haver um rio entre elas. Por outro lado, se houver uma estrada de fácil acesso passando por um local, até plantios não muito próximos poderiam fazer parte do mesmo grupo.
- Rede de transporte – o ponto de escoamento da madeira e as condições desse caminho é importante para saber se é viável ou não que a madeira saia por ele. Não só o caminho próximo do plantio, mas até as opções de via de transporte disponíveis entre o plantio e a fábrica.
- Área de corte – se, por algum motivo, a empresa precisar controlar a área total cortada, um limite deve ser estabelecido para os grupos de corte. Por exemplo, para não deixar grandes áreas desmatadas.
- Densidade da madeira – para a produção de celulose é muito importante o controle da densidade da madeira que está sendo processada. Não há grandes restrições, mas deve-se buscar um equilíbrio do total processado. Portanto, saber que um grupo de plantios fornecerá madeira de uma certa densidade num determinado período é essencial.

- Regeneração – vários custos envolvem o problema de organizar um novo plantio após a colheita. Isso implica em saber se devem ser plantadas novas mudas ou deixar que haja uma rebrota.
- Produtividade – a influência genética e o uso do solo podem fazer com as árvores se desenvolvam num ritmo diferente, podendo-se optar por cortar a madeira mais cedo, seja por alcançar um volume de corte mais rapidamente ou pelo custo de se esperar pelo seu desenvolvimento ser muito alto sendo melhor cortá-la para dar lugar a um novo plantio.
- Outros plantios – muitas vezes a empresa tem áreas com madeira não própria para a celulose que pode ser usada serraria pode até ser área nativa de preservação, portanto, devendo ser excluída do processo.

Existem ainda muitos outros fatores considerados para o planejamento de áreas florestais que envolvem desde a compra de novas áreas (e aonde comprá-las), o tipo de árvore a plantar (e o desenvolvimento de espécies melhores), cuidados no manejo e transporte até chegar no suprimento da fábrica. Sem esquecer das necessidades do mercado, que vão regular a produção.

Tomando o mapa da área total plantada e suas características, como ordenar o corte delas no tempo mais propício para suprir a fábrica? Este problema de agrupamento ou *clustering*, onde serão criados grupos ou unidades de corte com madeira homogênea, deve ter uma solução que oriente o analista para fazer um planejamento mais correto.

Neste estudo não temos a intenção de cobrir todas as variáveis que envolvem o problema, mas propor a adaptação de soluções existentes para agrupamento para o caso dos plantios. Serão escolhidas algumas características próprias dos plantios existentes para análise. Veja que há uma dificuldade muito grande de se mapear a rede de transporte e obstáculos naturais das áreas florestais sem o auxílio de um sistema de informação geográfica, levando-nos a considerar, neste trabalho, apenas a localização por latitude e longitude. No Capítulo 4 será definida a estratégia de solução com as características escolhidas para a análise. A identificação correta delas é a chave de entrada para um bom processamento das abordagens apresentadas no próximo capítulo.

## Capítulo 3 Ferramentas de Agrupamento

Agrupamento ou *clustering*, como visto, é a atividade de se criar grupos a partir de um conjunto diverso de elementos. Deve-se considerar ainda que não há interseção entre os grupos e que a união de todos os grupos é exatamente o conjunto inicial. Cada grupo é chamado de *cluster* e o resultado final é uma *clustering solution*.

Esse é o objetivo da Análise de *Cluster*, que pode ser encontrada na maioria das ciências naturais, sociais e da vida, discutida em estatística, matemática e computação, de uma literatura rica e diversa.

Para se definir regras de agrupamento é necessário caracterizar cada elemento. Depois, a análise é feita sobre dois tipos de entrada, basicamente. Pode ser cada elemento e seus dados em uma série de características ou a diferença entre cada elemento usualmente calculada em função das características de cada um.

Assim, dada uma das duas entradas, o objetivo é particionar o conjunto de elementos de forma que os grupos ou *clusters* resultantes sejam homogêneos e bem separados, ou seja, os elementos de um mesmo *cluster* devem ser o mais similar possível entre si e os elementos de *clusters* diferentes devem ser o mais dissimilar possível.

Qual o critério usar? Homogeneidade ou separação? Qual tipo de *clustering* considerar? Qual a complexidade da execução desse *clustering*? Como ele pode ser realizado? Como saber se os resultados do *clustering* são bons?

Essas considerações (Hansen e Jaumard, 1997) devem ser feitas ao se lidar com um problema de análise de cluster e, como se pode imaginar, existem diferentes abordagens, mas que podem ser agrupadas da seguinte forma (Michailidis, 2000):

- Métodos hierárquicos
- Métodos de Partição
- Métodos de teoria dos grafos

- Outras abordagens

Cada abordagem possui seu mérito e pode ser mais bem aplicada dependendo do problema.

### 3.1. Métodos Hierárquicos

Os métodos hierárquicos baseiam-se na iteratividade para criação dos grupos. Inicia-se com um elemento e, a cada passo da execução, um outro elemento ainda não agrupado é adicionado ao grupo já formado. Para definir qual será o novo elemento, todos os elementos não ainda agrupados são comparados com o grupo e o mais similar é escolhido.

A chave então, está na forma de executar essa comparação.

A visualização da formação dos grupos é muito bem vista através de um diagrama conhecido como *dendrogram* (Figura 2), ou simplesmente diagrama em árvore. Esse diagrama é uma árvore onde as folhas representam os elementos avaliados e as raízes representam grupos formados pelas suas ramificações. As arestas unem os grupos e indicam um peso para essa união, ou seja, a dissimilaridade entre cada grupo.

Observe que na Figura 2, os elementos 3 e 4 formam um grupo e a aresta que os une tem peso 5, No final da execução este grupo se une ao grupo maior com uma aresta de peso 11 aproximadamente.

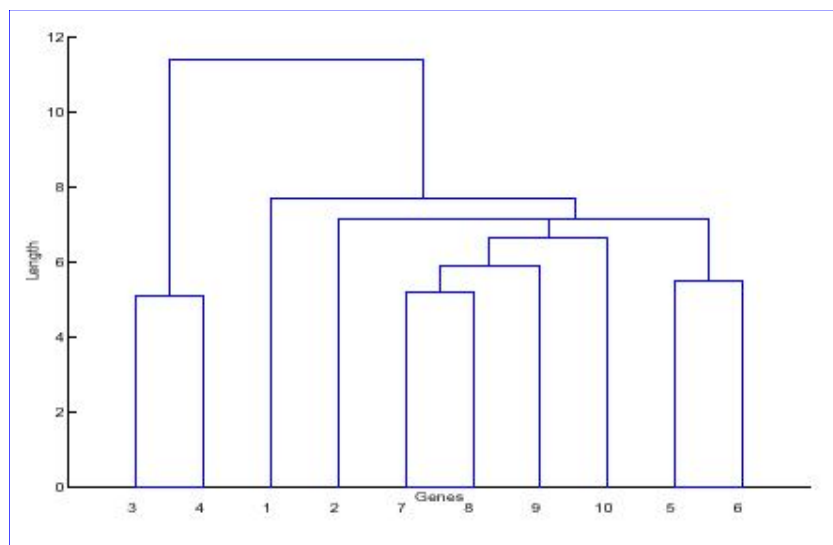


Figura 2- Diagrama em árvore

Para identificar a dissimilaridade ou diferença entre dois elementos deve-se encontrar o menor subconjunto que os contenha. Por exemplo, a diferença entre os elementos 2 e 10 é a aresta que une o *cluster* {2} ao *cluster* {7,8,9,10}, que tem peso 7. Quanto menor esse peso, maior a similaridade entre grupos.

Alguns autores fazem uso da seguinte desigualdade originada pelo processo:

Seja  $l_{ij}$  a diferença entre  $i$  e  $j$  e  $O$  o conjunto de *clusters* na árvore,

$$l_{ij} \leq \max \{l_{ik}, l_{jk}\}, i, j, k \in O \quad (1)$$

Neste caso, a matriz de dissimilaridade seria transformada em uma matriz de distâncias.

Entretanto, uma atenção maior é dada não a essas distâncias, mas sim à ordenação dos tamanhos, através de algoritmos aglomerativos e uma visão *bottom-up* para juntar repetidamente conjuntos de objetos, como na figura anterior.

A idéia é tornar cada elemento um *cluster* e a partir daí começar a avaliar qual o par de *clusters* mais similar. Esse par é transformado num novo *cluster* e o processo se repete até que tenhamos um único *cluster* contendo todos os elementos.

Pelas definições apresentadas por Michailidis, existem três formas principais para se avaliar a similaridade entre *clusters* por métodos aglomerativos, lembrando que o valor associado na matriz é a dissimilaridade: *single linkage*, *complete linkage* e *average linkage*.

Sejam  $C_i$  e  $C_j$  *clusters* |  $C_i \in O$  e  $C_j \in O$ ,

$d(s,t)$  a dissimilaridade entre dois elementos de  $O$  e

$n_k$  o número de elementos do *cluster*  $k$ ,

a dissimilaridade  $d(C_i, C_j)$  entre dois *clusters* de  $O$  é definida por:

$$\text{Single Linkage: } d(C_i, C_j) = \min_{s,t} \{d(s,t) \mid s \in C_i, t \in C_j\} \quad (2)$$

$$\text{Complete Linkage: } d(C_i, C_j) = \max_{s,t} \{d(s,t) \mid s \in C_i, t \in C_j\} \quad (3)$$

$$\text{Average Linkage: } d(C_i, C_j) = \frac{\sum_{s \in C_i} \sum_{t \in C_j} d(s,t)}{n_i n_j} \quad (4)$$

Quando as diferenças entre *clusters* forem iguais ou muito próximas, não se tem um par mais similar muito bem definido e a escolha pode influenciar muito os resultados, o que não permite, para a maioria dos critérios de avaliação, assumir que a árvore completa satisfaz algum critério de otimalidade.

Vejam os um exemplo de uso do *single linkage*:

Seja  $O$  um conjunto de 5 elementos com a seguinte matriz de dissimilaridade:

$$D = \begin{bmatrix} 0 & & & & \\ 15 & 0 & & & \\ 4 & 8 & 0 & & \\ 3 & 5 & 10 & 0 & \\ 11 & 4 & 9 & 6 & 0 \end{bmatrix}, \text{ onde } d(i,j) \text{ é a dissimilaridade entre } i \text{ e } j, \text{ com } i,j \in \{0,1,2,3,4\}$$

Inicialmente, cada elemento é transformado em um *cluster*. A menor dissimilaridade da matriz é 3 e ocorre entre os *clusters* 3 e 0. Os dois se unem e a dissimilaridade entre o novo *cluster* e os restantes deve ser calculada da forma apresentada em (2). Veja:

$$d((3,0),1) = \min\{d(3,1),d(0,1)\} = \min\{5,15\} = 5$$

$$d((3,0),2) = \min\{d(3,2),d(0,2)\} = \min\{10,4\} = 4$$

$$d((3,0),4) = \min\{d(3,4),d(0,4)\} = \min\{6,11\} = 6$$

O novo *cluster* é inserido na matriz e os *clusters* 3 e 0 são retirados. Assim, temos uma nova matriz:

$$D = \begin{bmatrix} 0 & & & \\ 5 & 0 & & \\ 4 & 8 & 0 & \\ 6 & 4 & 9 & 0 \end{bmatrix} \begin{matrix} < -\{3,0\} \\ < -\{1\} \\ < -\{2\} \\ < -\{4\} \end{matrix}$$

Agora, a menor dissimilaridade é 4 e ocorre em dois casos (2,0) e (3,1). Qual dos dois escolher?

Sem muita complexidade, pode-se escolher o primeiro e então temos:

$$d((2,0),1) = \min\{d(2,1),d(0,1)\} = \min\{8,5\} = 5$$

$$d((2,0),3) = \min\{d(2,3),d(0,3)\} = \min\{9,6\} = 6$$

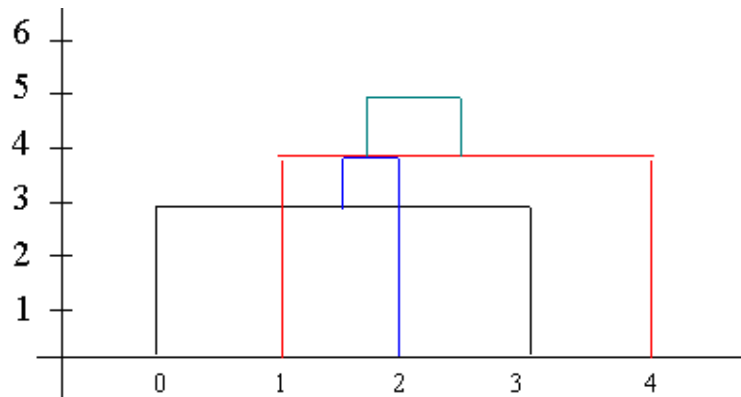
O novo *cluster* é inserido na matriz e os *clusters* 2 e 0 são retirados. Assim, temos uma nova matriz:

$$D = \begin{bmatrix} 0 & & \\ 5 & 0 & \\ 6 & 4 & 0 \end{bmatrix} \begin{array}{l} < -\{2,3,0\} \\ < -\{1\} \\ < -\{4\} \end{array}$$

A próxima união será entre 1 e 2 da nova matriz, fazendo

$$d((1,2),0) = \min\{d(1,0),d(2,0)\} = \min\{5,6\} = 5$$

A Figura 3 mostra o diagrama resultante:



**Figura 3 -Diagrama por *single linkage***

Se no segundo passo fossem escolhidos os *clusters* 3 e 1, teríamos:

$$d((3,1),0) = \min\{d(3,0),d(1,0)\} = \min\{6,5\} = 5$$

$$d((3,1),2) = \min\{d(3,2),d(1,2)\} = \min\{9,8\} = 8$$

O novo *cluster* é inserido na matriz e os *clusters* 2 e 0 são retirados. Assim, temos uma nova matriz:

$$D = \begin{bmatrix} 0 & & \\ 5 & 0 & \\ 4 & 8 & 0 \end{bmatrix} \begin{array}{l} < -\{3,0\} \\ < -\{1,4\} \\ < -\{2\} \end{array}$$

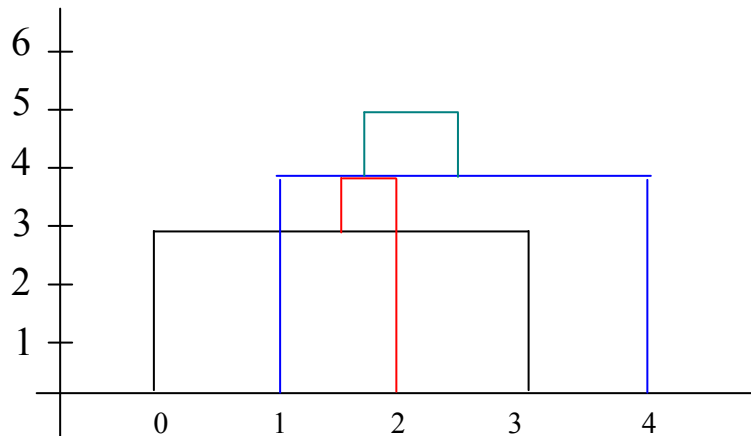
A próxima união será entre 0 e 2 da nova matriz, fazendo

$$d((0,2),1) = \min\{d(0,1),d(2,1)\} = \min\{5,8\} = 5$$

Por fim, o diagrama resultante é apresentado na Figura 4.

Nesse exemplo o resultado final foi o mesmo, só mudando a ordem de geração dos *clusters*, mas de uma forma geral a técnica pode não discernir *clusters* mal separados.

Os outros tratamentos podem produzir diferentes resultados também, principalmente nos níveis mais baixos, por causa da ordem de escolha das uniões.



**Figura 4 - Diagrama com segunda escolha**

### **3.2. Métodos de Partição**

Os métodos hierárquicos vistos anteriormente fornecem uma configuração de clusters de forma seqüencial, de forma que a solução pode ser escolhida dentro dessa configuração, avaliando-se o peso das arestas que uniram *clusters* intermediários. Afinal, o último passo forma um único *cluster*. A abordagem dos métodos de partição chega a um conjunto final de *clusters* baseada na informação de quantos *clusters* devem ser gerados. O conjunto de elementos é avaliado para que nos  $K$  *clusters* necessários, existam elementos bem semelhantes e entre os diferentes *clusters* haja uma boa separação. Uma questão que algoritmos dessa abordagem devem definir, portanto, é se priorizam a avaliação da homogeneidade ou da separação dos *clusters*.

Com base nas informações de Michailidis, pode-se trabalhar com a idéia de diâmetro de um cluster (a dissimilaridade máxima entre seus elementos), com a soma das dissimilaridades entre elementos do *cluster* e os que estão fora dele (separação), ou mesmo a menor dessas dissimilaridades, entre outras.



Caso não queira priorizar uma das avaliações pode-se encontrar um equilíbrio entre separação e homogeneidade, o importante é procurar um resultado ótimo. Para particionar  $n$  elementos em  $K$  *clusters* não vazios existem  $N(n,K)$  formas, dadas por:

$$N(n, K) = \frac{1}{K!} \sum_{k=1}^K (-1)^{K-k} \binom{K}{k} k^n \quad (5)$$

Dessa forma, a maioria dos algoritmos conhecidos precisa de um número polinomial  $n$  de operações - são problemas conhecidos por Nphard.

Um algoritmo de partição bem conhecido é o K-Means (MacQueen, 1965). Sua idéia é ter um número  $K$  de partições ou *clusters* no início do processamento e, a partir daí, começar a avaliar uma série de transformações possíveis para que elementos de uma partição migrem para outra. Cada elemento é comparado com os *clusters* diferentes do seu atual e se o custo desse outro *cluster* com o elemento for menor do que o custo dele sem o elemento, então o elemento passa a compor o *cluster*.

Seja  $N$  um conjunto de elementos,  $P$  um conjunto de  $K$  partições de  $N$  e  $C_p^{ij}$  o custo da solução  $P$  considerando que o elemento  $i$  seja movido para o *cluster*  $j$ .

1- Inicie um conjunto  $P$  de partições de  $N$  em  $K$  *clusters*

2 – Enquanto existir elemento  $i \in N$  e *cluster*  $j \neq P(i)$  tal que  $\text{mínimo}(C_p^{ij}) < C_p$

Mova  $i$  para  $j$

**Figura 5 – Algoritmo K-Means**

O custo  $C_p$  na Figura 5 é originalmente calculado por MacQueen como sendo o somatório das distâncias euclidianas entre o vetor de características (*fingerprints*) de cada elemento  $i$  de  $P$  com o vetor da centróide do *cluster* de  $i$ . Assim, o algoritmo busca minimizar a dissimilaridade entre os componentes dos *clusters*.

Para contornar a restrição de se determinar o número de *clusters* a gerar, Herwig (Herwig et ali, 1999) apresenta uma adaptação ao algoritmo anterior onde é necessário informar o similaridade máxima admitida entre dois *clusters* distintos e também entre um elemento e um *cluster* diferente do seu. Esses dois parâmetros são denotados respectivamente por  $\gamma$  e  $\rho$ .

O algoritmo (Figura 6) parte de um conjunto inicial de *clusters* e então, cada elemento ainda fora de *clusters* é comparado com os *clusters* existentes. Se sua similaridade ao *cluster* for maior que  $\rho$ , adiciona-se este elemento ao *cluster*. Se nenhum *cluster* for suficientemente similar, forma-se um novo *cluster*. À medida que as inserções acontecem, os *clusters* também são comparados entre si para identificar se algum par de *clusters* tem similaridade maior que  $\gamma$  para uni-los. Um detalhe importante é que um elemento pode ser adicionado a mais de um *cluster* e isso influencia no cálculo da centróide do *cluster* e portanto, nas comparações seguintes. Por isso, ao final do processo, cada elemento fica associado ao *cluster* mais similar.

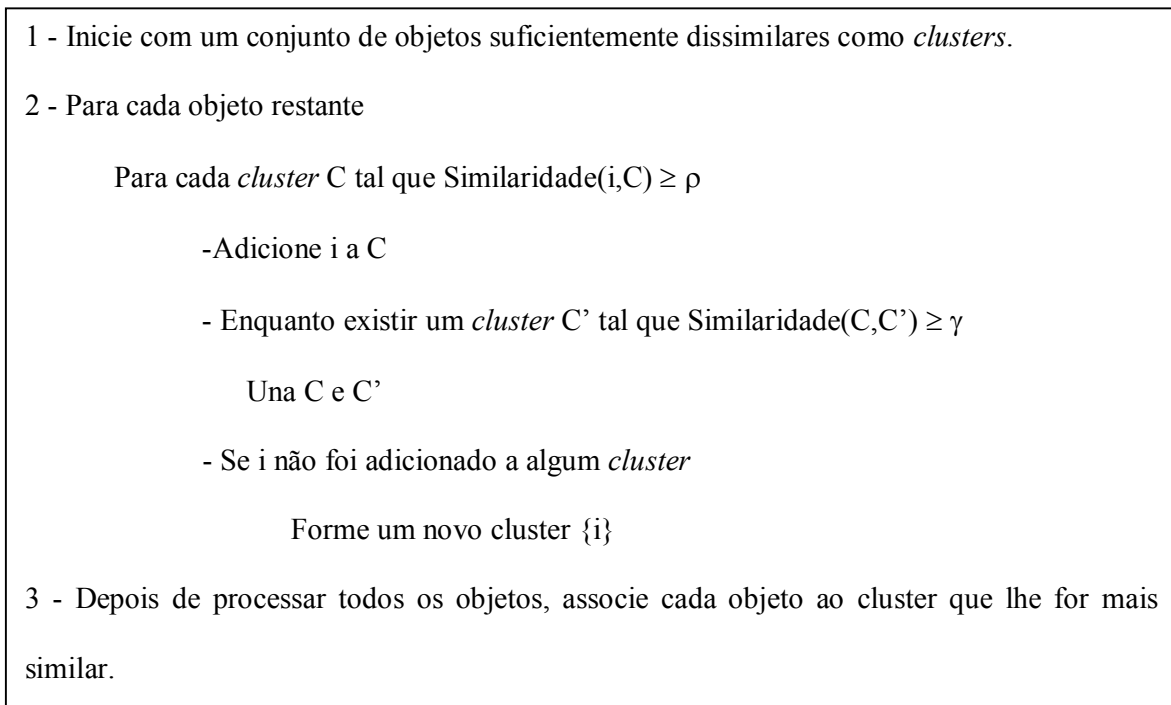


Figura 6 – Algoritmo de Herwig et ali

### 3.3. Métodos com abordagem em teoria dos grafos

Nessa abordagem a matriz de similaridade é mapeada em uma estrutura de grafo, onde cada vértice representa um elemento e cada aresta indica a similaridade entre os elementos que conecta.

Antes de entrar nas abordagens, vamos a algumas definições importantes sobre grafos:

- $G(V,E)$  é um grafo com um conjunto  $V$  de vértices e  $E$  de arestas.

- $G_R = G(R, E_R)$  é o subgrafo induzido por um subconjunto  $R \subseteq V$  de vértices, retirando-se todos os vértices que não estão em  $R$  e todas as arestas incidentes nestes, e com  $E_R = \{(i,j) \in E : i,j \in R\}$
- Um corte (ou *cut*)  $T$  é um subconjunto de arestas cuja remoção desconecta  $G$ .
- $|G|$  é o número de vértices de  $G$ .
- $|T|$  é o número de arestas de  $T$ .
- O peso de  $T$  é a soma dos pesos de suas arestas
- Um corte de peso mínimo de  $G$  ou (*minimum weighted cut*) é um corte  $T$  de  $G$  com peso mínimo.
- O caminho entre dois vértices  $u$  e  $v$  é o conjunto de arestas que ligam  $u$  a  $v$  em  $G$ .
- A distância  $d(u,v)$  entre os vértices  $u$  e  $v$ , é o tamanho do menor caminho entre  $u$  e  $v$ . Se não existe caminho,  $d(u,v) = \infty$ .
- O diâmetro  $diam(G)$  é a distância máxima entre qualquer par de  $G$ .
- O grau  $deg(v)$  é o número de arestas incidentes em  $v$ .
- Clique é um grafo completo, ou seja, qualquer vértice está conectado a qualquer outro vértice.

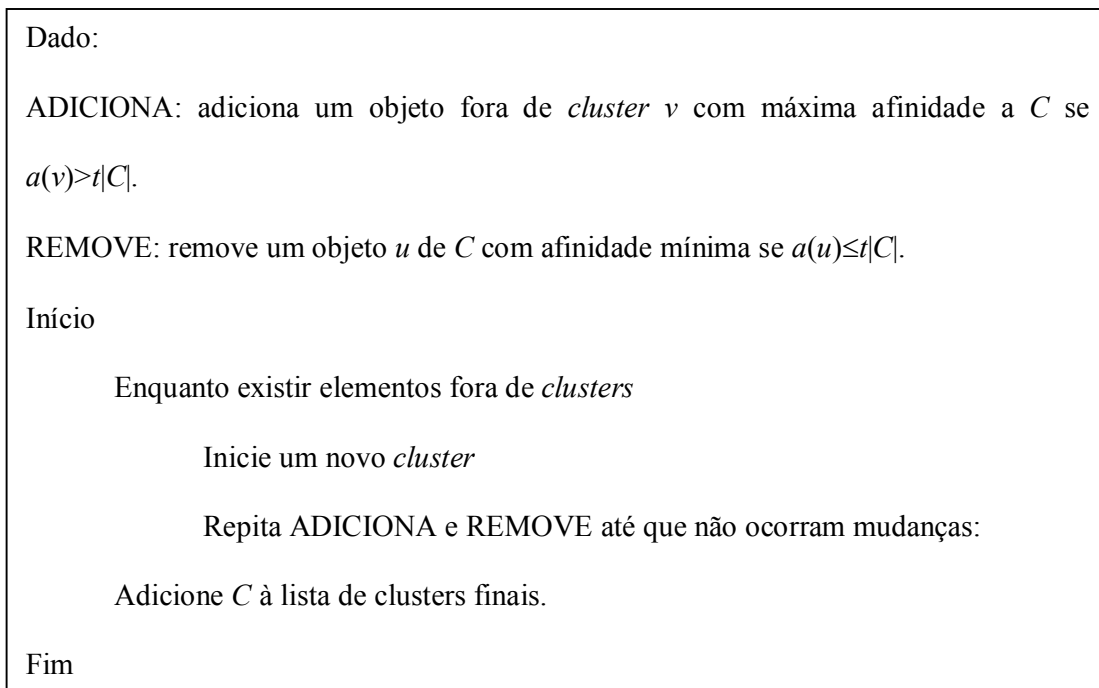
O conjunto de *clusters* da solução é alcançado realizando-se cortes no grafo de forma que cada subgrafo tenha uma similaridade ótima entre seus vértices (arestas de peso máximo) e uma ótima separação entre vértices de subgrafos distintos (arestas de peso mínimo). Não deve haver interseção entre os subgrafos.

Procurando-se por valores mínimos de similaridade entre subgrafos diferentes, o conjunto de arestas que gera o corte deve ter peso mínimo.

Um primeiro algoritmo com essa abordagem é o CAST (*Cluster Affinity Search Technique*), introduzido por Ben-Dor (Ben-Dor et al, 1999), que trabalha com uma matriz de similaridade onde  $Similaridade(i,j) \in [0,1]$ . Para as avaliações de similaridade é assumido um marco (*threshold*) que indica um limite mínimo de similaridade para que um vértice esteja em um *cluster*. Essa similaridade é chamada de afinidade ( $a$ ) e a afinidade de  $v$  a um *cluster*  $C$  é dada por:

$$a(v) = \sum_{u \in C} \text{Similaridade}(u, v) \quad (6)$$

Assim, o algoritmo (Figura 7) inicia com um elemento como cluster e começa a avaliar sua afinidade com os elementos restantes, adicionando-os se ela for maior do que o *threshold*. À medida que elementos novos são adicionados, os antigos devem ter sua afinidade reavaliada, pois se for menor que o *threshold*, deve ser retirado do cluster. Quando nenhum elemento for adicionado, reinicia-se o processo com um elemento fora de *clusters* como um novo *cluster*. Os *clusters* vão sendo gerados um a um até que todo o processo se estabilize. Com isso, o algoritmo pode construir uma solução melhor do que a dos algoritmos hierárquicos, já que uma decisão pode ser reavaliada num passo seguinte.



**Figura 7 – Algoritmo CAST**

Uma outra abordagem é o HCS (*Highly Connected Subgraph*) definido por Hartuv e Shamir (Hartuv e Shamir, 2000) que, semelhante ao CAST, tem um marco (*threshold*) pré-definido para verificar quais vértices têm similaridade maior do que o *threshold* e definir o peso das arestas que os conectam. Esse peso tem valor 1 para todas as arestas entre vértices com similaridade maior que o *threshold* e valor 0, ou sem aresta, para vértices não conectados por baixa similaridade.

Um grafo  $G$  é altamente conectado se o seu valor de corte mínimo excede à  $\frac{|G|}{2}$ , ou seja,  $G$  permanece conectado mesmo que  $\frac{|G|}{2}$  arestas sejam removidas.

O algoritmo busca então, particionar um grafo em subgrafos que sejam altamente conectados, chamados *kernels*. A homogeneidade dos *clusters* gerados é muito forte e comprovada pelos estudos do autor, pois o diâmetro de cada *cluster* gerado é de no máximo 2 e o número de arestas do *cluster* é maior ou igual à metade do número de arestas de um suposto clique para o *cluster*.

Por último, existe uma abordagem que utiliza *minimum spanning trees (MST)* no algoritmo de *clustering*. É uma abordagem prática e interessante apresentada por Tavakoly (Tavakoly, 2003), originalmente para tratamento de informações de genes.

Cada cluster corresponde a uma subárvore da MST, assim um problema de *clustering* multidimensional é convertido num problema de particionamento de árvore, trazendo a vantagem da estrutura simples de uma árvore, que facilita implementações eficientes e de baixo custo computacional. Além disso, não depende de uma forma geométrica detalhada do cluster, o que elimina muitas das dificuldades encontradas em algoritmos como o *K-means* e um outro conhecido por *Self Organized Map* (Kohonen, 1990), baseado em redes neurais, que não abordamos aqui.

Ainda existem outras abordagens, mas não entramos em detalhes por achar melhor aprofundar os estudos sobre HCS, que nos pareceu um método muito satisfatório para o problema dos plantios e sobre uso de *MST*, que é bem prático, para simular a atuação humana. Por isso, nos dois próximos capítulos faremos um detalhamento sobre estes algoritmos.

### 3.3.1 Algoritmo HCS (*highly connected subgraph*)

Como apresentado no capítulo anterior, o algoritmo HCS usa um *threshold* pré-definido para criar arestas entre os vértices do grafo que tenham similaridade maior que o *threshold*. Para chegar a um conjunto de subgrafos altamente conectados Shamir e Sharan apresentam um algoritmo básico que avalia se o subgrafo correntemente tratado é um *kernel* (altamente conectado). Se não for, um corte

de peso mínimo é calculado e os dois novos subgrafos resultantes recebem o mesmo tratamento de avaliação. Quando ocorrer um subgrafo com apenas um vértice, ele é adicionado a um conjunto de *singletons* (objetos isolados). Quando não ocorrerem mais cortes, o algoritmo retorna uma lista de *kernels* que são base para eventuais *clusters*.

O algoritmo básico para esse processo é mostrado na Figura 8 abaixo.

```
Se  $V(G)=\{v\}$  //tem um único elemento
    mova  $v$  para o conjunto de singletons  $R$ .
Senão se  $G$  for um Kernel
    retorne  $V(G)$ .
Senão
     $(H, \bar{H}) \leftarrow \text{MinWeightedCut}(G)$ 
     $\text{FormKernels}(H)$ 
     $\text{FormKernels}(\bar{H})$ 
```

**Figura 8 – Algoritmo *FormKernels***

Além do HCS, seus autores apresentam outro algoritmo com essa mesma base de processamento chamado CLICK (*Cluster Identification via Connectivity Kernels*). Antes de aprofundarmos no HCS, vamos comentar um pouco sobre este algoritmo.

CLICK usa conceitos de probabilidade para verificar se o subgrafo é um *kernel*. Ele será um *kernel* quando para todo corte  $T$  no subgrafo, a probabilidade dele conter somente arestas entre pares for maior do que a probabilidade de  $T$  conter somente arestas entre não-pares. Também se assume que a similaridade de um par de vértices é distribuída normalmente (por uma função de densidade normal) e calcula-se a probabilidade de que dois vértices escolhidos aleatoriamente sejam pares. Com isso, o grafo de similaridade formado tem no peso  $w_{ij}$  da aresta  $(i,j)$  um reflexo da probabilidade de  $i$  e  $j$  serem pares.

Na Figura 9 há uma apresentação do algoritmo. Primeiro são produzidos *kernels*. Depois, nessa base de eventuais *clusters*, são adicionados (ou adotados) os *singletons*, verificando a similaridade entre as características (*fingerprints*) deles com os *clusters*. Ainda se realiza uma união entre *kernels* com similaridade maior do que o *threshold* e, por último mais uma adoção para os *singletons* restantes.

```
Dado  $N$ , o conjunto inicial de elementos no grafo de similaridade
 $L$ , a lista corrente de kernels
 $R$ , o conjunto de singletons e
 $G_R$ , o subgrafo induzido por  $R$ 
Início
     $R \leftarrow N$ 
    Enquanto alguma mudança ocorrer
         $Form-Kernels(G_R)$ 
        Adoção( $L', R$ )
        União( $L', R$ )
        Adoção( $L', R$ )
Fim
```

**Figura 9 – Algoritmo CLICK**

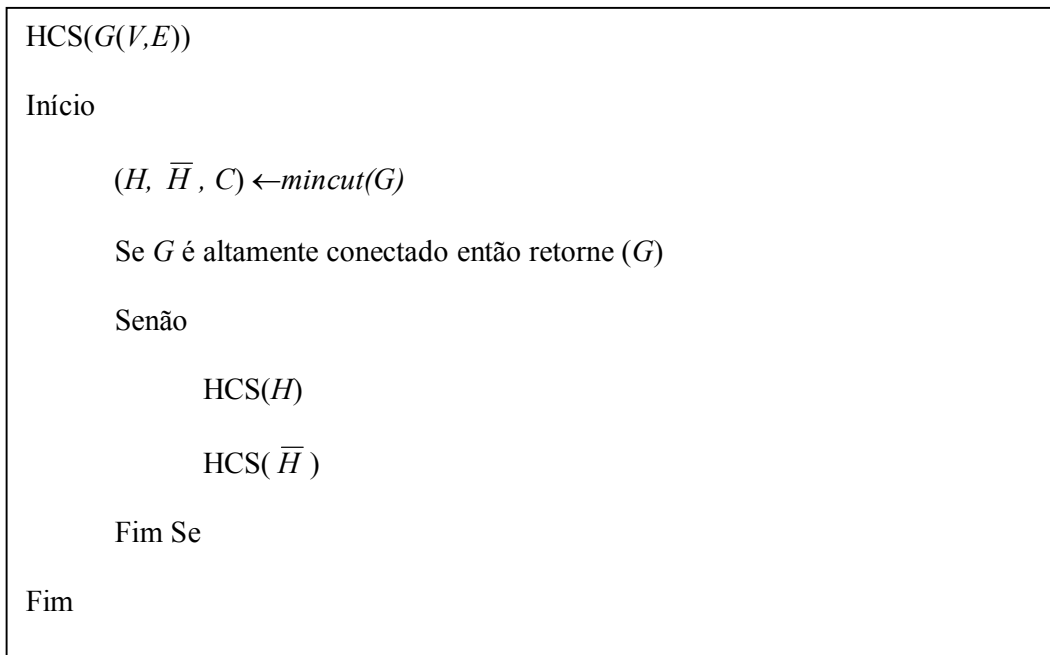
Para reduzir o tempo de execução do algoritmo em grandes instâncias uma heurística de recusa é aplicada, calculando-se o peso médio dos vértices proporcional ao tamanho do conjunto e comparando essa média com o peso dos vértices, para que vértices de pouco peso sejam retirados de grandes conjuntos. À medida que vértices sejam retirados, um novo peso médio é calculado. Os vértices removidos são marcados como *singletons* e tratados após o processamento do algoritmo.

O HCS (Figura 10) também se baseia no algoritmo *Formkernels* e para descrever o algoritmo precisamos de algumas definições a mais:

- $k(G)$  é o mínimo  $k$  de arestas cuja remoção resulta num grafo desconexo. Se  $k(G)=l$ ,  $G$  é um grafo  $l$ -conexo.
- $S$  é um corte mínimo (*mincut*) de um grafo não-trivial  $G$  se e somente se  $|S|=k(G)$ .

Os subgrafos altamente conectados que o algoritmo busca devem ser como um subgrafo  $H \subseteq G$  com

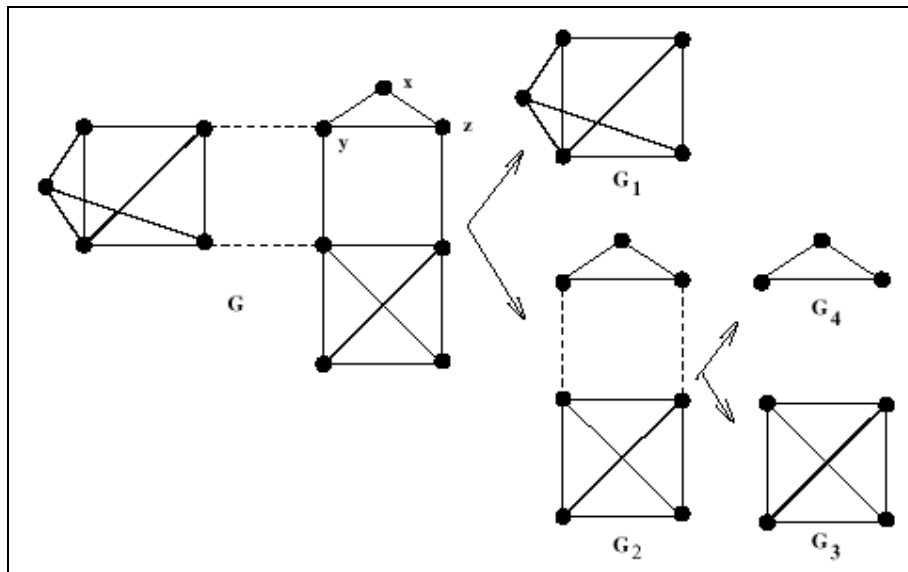
$k(H) = \frac{|H|}{2}$ . Estes subgrafos são identificados como *clusters*.



**Figura 10 – Algoritmo HCS**

Segue abaixo (Figura 11) um exemplo citado por Hartuv a Shamir, onde o HCS recebe um grafo  $G$  e o divide em  $G_1$  e  $G_2$ . Depois, divide  $G_2$  em  $G_3$  e  $G_4$ , dando como resultado final a solução  $\{G_1, G_3, G_4\}$ . Os cortes realizados estão representados pelas linhas pontilhadas.





**Figura 11 – Exemplo de execução do HCS**

Dado um grafo  $G$  com  $n$  vértices e  $m$  arestas, o número  $N$  de *clusters* encontrado e a complexidade  $(n,m)$  de se computar um corte mínimo no grafo  $G$ :

O tempo de execução do HCS é no máximo  $2N x (n,m)$ .

De acordo com Hartuv e Shamir, o HCS tem as seguintes propriedades:

1. O diâmetro de qualquer grafo altamente conectado é dois

Considerando um grafo  $G$ , um corte mínimo  $S$  em  $G$  e os subgrafos induzidos  $H$  e  $\bar{H}$  gerados por  $S$ :

2. Se  $\bar{H}$  é o menor subgrafo e  $|\bar{H}| > 1$ , então  $|S| \leq |\bar{H}|$ , com igualdade somente se  $\bar{H}$  for um clique.
3. Suponha que  $G$  não seja altamente conectado, mas tenha diâmetro 2 e  $|\bar{H}| \leq |H|$ . Então:
  - a. Cada vértice em  $\bar{H}$  é incidente em  $S$ .
  - b.  $\bar{H}$  é um clique.
  - c. Se  $|\bar{H}| > 1$ , cada vértice em  $\bar{H}$  incide numa única aresta de  $S$ .
4. O número de arestas num grafo altamente conectado é quadrático.
5. O número de arestas removidas em cada iteração do HCS é no máximo linear.

A forte homogeneidade é alcançada pelas propriedades 1 e 4, que indicam que no máximo duas arestas separam qualquer par de vértices de um grafo altamente conectado e que ele tem pelo menos a metade da densidade de um clique.

Já a forte separação entre *clusters* é alcançada com as propriedades de 3 e 5, pois qualquer conjunto não trivial, dividido pelo algoritmo dificilmente terá diâmetro 2 e o número de arestas removidas a cada iteração é linear em comparação com o número quadrático dos *clusters* finais.

Para incrementar a velocidade e a performance do algoritmo na prática, existem algumas sugestões de Hartuv e Shamir:

- Repetir o HCS um certo número de vezes para evitar, no caso de haver mais de um corte mínimo possível, que a escolha de um corte mínimo não leve à melhor solução e até gere *singletons* desnecessários, como na Figura 11 se escolher um corte que tire  $x$  de  $G_2$ . Na prática poucas iterações são necessárias (1 a 5).
- Executar o processo “adoção” de *singletons* para que cada *singleton* seja adicionado a um *cluster* com o qual tenha um número suficientemente grande de vizinhos. Como isso altera a configuração atual, o processo deve ser repetido para acomodar as adoções feitas.
- Remover vértices de baixo grau para diminuir a probabilidade de se realizar um corte que gera *singletons*, pois é custoso e pouco informativo para o agrupamento, além de poder acontecer várias vezes num grafo extenso. Com isso, o tempo de execução reduz.

Com os refinamentos incluídos, temos um novo algoritmo (Figura 12), onde  $d_1, d_2, \dots, d_p$  é uma seqüência decrescente de inteiros fornecida como entrada externa para validação do grau dos vértices:

HCS\_LOOP ( $G(V,E)$ )

Início

Para ( $i=1$  até  $p$ ) faça

    Remova vértices já em clusters de  $G$

$H \leftarrow G$

    Repedidamente remova todos os vértices de grau  $< d(i)$  de  $H$

    Até Que (nenhum cluster seja encontrado pela chamada do HCS) faça

        HCS( $H$ )

        Execute adoção de *singletons*

        Remova vértices já em clusters de  $H$

    Fim Até Que

Fim Para

Fim

**Figura 12 – Algoritmo HCS incrementado**

Pelas informações de Hartuv e Shamir, a execução do algoritmo tanto com dados simulados quanto reais teve resultados muito bons, concluindo que:

- HCS tem complexidade polinomial baixa.
- É eficiente na prática.
- Não é necessário predefinir o número de clusters.
- Gera clusters de diâmetro 2 - homogeneidade.
- Pode ser interessante não assumir como *cluster* um subgrafo  $H$  com conectividade  $\frac{n}{2}$ , mesmo

com diâmetro 2, pois consistem de dois cliques com  $\frac{n}{2}$  vértices conectados por um casamento perfeito (*perfect matching*). Ao menos que  $n$  seja muito pequeno, isso parece mais uma união de dois clusters.

- Pode-se também tratar subgrafos com  $n = 2$  de forma especial nas implementações.
- Para o futuro do algoritmo, eles propõem trabalhar com peso nas arestas e ver a possibilidade de determinar um valor ótimo para o marco (*threshold*) dos valores de similaridade usados para formar o grafo de similaridade.

Para implementar o algoritmo, é necessária uma solução para o problema de corte mínimo. Uma solução é apresentada a seguir.

### 3.3.1.1. Algoritmo de Corte Mínimo

Dados  $G=(V,E)$ , um grafo não direcionado

$u_e$ , uma função de capacidade, onde  $u_e > 0$  para cada  $e \in E$ ,

$\delta(S) = \{vw: vw \in E, v \in S, w \notin S\}$ , onde  $S \subset V$ ,

O problema do corte mínimo se resume em encontrar um conjunto  $A = \delta(S)$  tal que  $S \subset V$  e  $u(A)$  seja mínimo.

Corte mínimo em grafos é um tema muito discutido nas teorias sobre problemas de fluxo máximo, onde se observa que a solução de um fluxo máximo é solução para um corte mínimo (Cook, W. J. et ali, 1998). Especificamente para grafos não direcionados, o problema de corte mínimo pode ser resolvido de forma mais eficiente do que com a aplicação de métodos de fluxo máximo.

Podemos assumir que  $G$  é conexo, pois caso contrário, a solução é trivial. Quando  $u_e = 1$  o mínimo é a conectividade das arestas de  $G$ , ou  $k(G)$  visto na seção 3.3.1. Como um corte  $\delta(S)$  separa  $v, w$  de  $G$  se e somente se exatamente um dos dois vértices está em  $S$ ,  $\delta(S)$  é um  $(v, w)$ -cut.

Cook apresenta um algoritmo para corte mínimo com base no que ele chama de “Identificação de Nós”. Sejam  $v, w$  vértices distintos de  $G(V, E)$ . Identificar um vértice  $v$  em  $w$  é retirar do grafo os dois vértices mais a aresta (ou as arestas) que os conecta e inserir um novo vértice  $x$ , gerando  $G_{vw}$ . Para cada aresta  $e \in E$ , o extremo  $p$  de  $e$  em  $G$ , será o extremo de  $e$  em  $G_{vw}$  se  $p \neq v, w$ . Caso contrário, o extremo  $p$  de  $e$  será o novo vértice  $x$ .

Esse processo é interessante pela observação de que todo corte de  $G_{vw}$  é um corte de  $G$  e todo corte de  $G$  que não separe  $v$  de  $w$  é um corte de  $G_{vw}$ .

Suponha que façamos uma seqüência de identificações de nós começando por  $G$ . Então, temos um grafo  $G'$ . Cada um desses cortes é um corte de  $G$ . Assim, um corte mínimo de  $G'$  é um candidato a ser um corte mínimo de  $G$ , mas alguns cortes de  $G$  foram "perdidos", ou seja, não são cortes de  $G'$ .

Isso indica que podemos resolver o problema de corte mínimo pelo seguinte processo: escolha um par de nós  $v,w$ , compute um  $(v,w)$ -cut mínimo em  $G$  e substitua  $G$  por  $G_{vw}$ . Depois de  $n-2$  passos, vamos ter um grafo para o qual o problema de corte mínimo é trivial, pois terá apenas um corte. Assim, temos acumulado  $n-1$  cortes em vários grafos obtidos de  $G$  por identificação de nós e o melhor dos cortes é um corte mínimo de  $G$ . Se fizermos isso numa seqüência direta, teremos que resolver  $n-2$  problemas de  $(r,s)$ -cut mínimos.

Como a escolha de  $v,w$  está ao nosso alcance, podemos encontrar um  $(v,w)$ -cut mínimo de forma simples buscando uma ordem legal (*legal ordering*) de  $G$ , que é uma ordenação  $v_1, v_2, \dots, v_n$  de vértices de  $G$  tal que

$$u(\delta(V_{i-1}) \cap \delta(v_i)) \geq u(\delta(V_{i-1}) \cap \delta(v_j)) \text{ para } 2 \leq i < j \leq n, \text{ onde } V_i = \{v_1, \dots, v_i\}. \quad (7)$$

Ou seja, escolhemos qualquer vértice para ser  $v_1$  e no passo  $i$  escolhemos  $v_i$  para ser o vértice com a maior capacidade total de arestas unindo-o aos vértices escolhidos anteriormente. Por exemplo, na Figura 13, uma ordem legal começando por  $a$  é  $a, b, c, d, e, h, g, f$ . Uma ordem legal pode ser encontrada em tempo  $O(n^2)$ .

Se  $v_1, \dots, v_n$  é uma ordem legal de  $G$ , então  $\delta(v_n)$  é um  $(v_n, v_{n-1})$ -cut mínimo de  $G$ .

A Figura 14 mostra o efeito da identificação dos nós  $f, g$  no exemplo da Figura 13 abaixo.

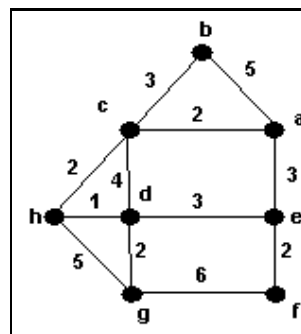
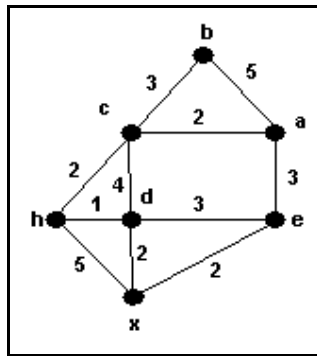


Figura 13 – Exemplo de entrada para o problema de corte mínimo



**Figura 14 – “Identificação” de nós**

De fato, no exemplo da Figura 13,  $\delta(f)$  é um  $(f,g)$ -cut mínimo. Portanto, um corte mínimo deste grafo é  $\delta(f)$  ou um corte mínimo do grafo da Figura 14. O procedimento de ordem legal indica que o algoritmo (Figura 15) abaixo retorna um corte mínimo e ainda executa num tempo  $O(n^3)$ .

```

Inicie  $M$  com  $\infty$  e  $A$  como indefinido;

Enquanto  $G$  tiver mais que um vértice
    Encontre uma ordem legal  $v_1, v_2, \dots, v_n$  de  $G$ ;
    Se  $u(\delta(v_n)) < M$ 
        Substitua  $M$  por  $u(\delta(v_n))$  e  $A$  por  $\delta(v_n)$ ;
    Substitua  $G$  por  $G_{v_n-1v_n}$ ;

Retorne  $A$ .

```

**Figura 15 - Algoritmo de Corte Mínimo por Identificação de Nós**

Segue um exemplo da execução do corte mínimo por identificação de nós:

Tomando a Figura 13, a execução do algoritmo tem os seguintes passos:

Identificar uma ordem legal para  $a, \dots, h$

Começando com o vértice  $a$ , a aresta de maior peso é  $b$ . Assim, unimos  $a$  e  $b$ , como na Figura 16.

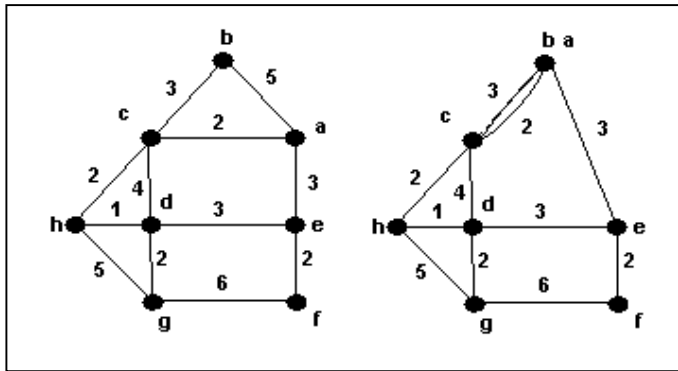


Figura 16 – Início do primeiro passo do *minimum cut*

Continuando por  $(a,b)$ , selecionamos a aresta de maior peso  $c$ , pois as duas arestas que surgiram somam os pesos e assim por diante, como mostra a Figura 17.

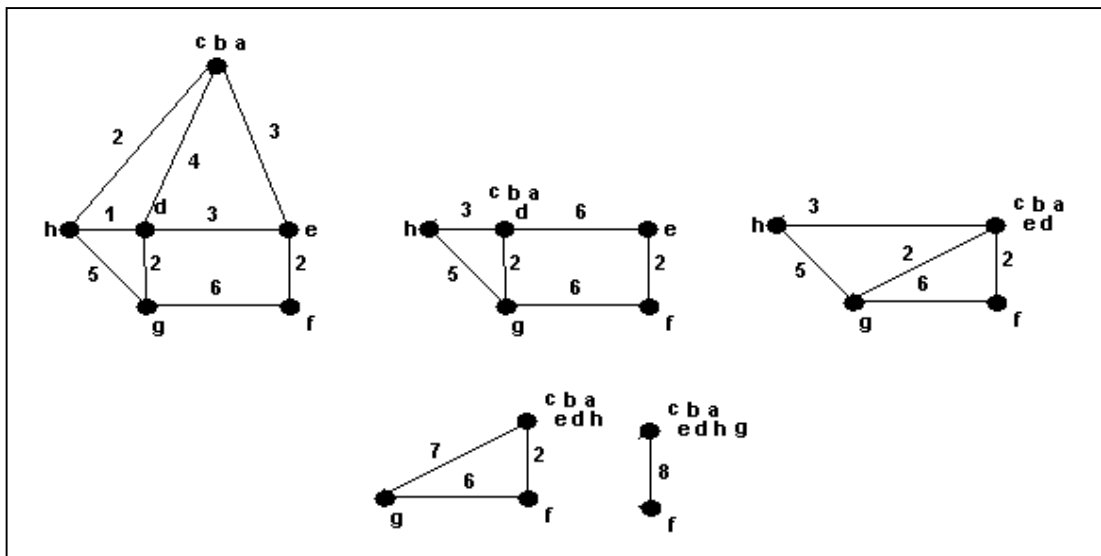


Figura 17 – Final do primeiro passo do *minimum cut*

Neste ponto então, temos uma ordem legal  $a,b,c,d,e,h,g,f$ , onde  $\delta(v_n) = \delta(f) = \{g,f\}$  e

$$u(\delta(v_n)) = u(\delta(f)) = 8.$$

$$M=8 \text{ e } A=\delta(f).$$

Para o segundo passo (Figura 18), substituímos  $v_n, v_{n-1}$  por um novo vértice  $x$ , substituímos  $G$  por  $G_{v_n - v_n}$  e reiniciamos o processo.

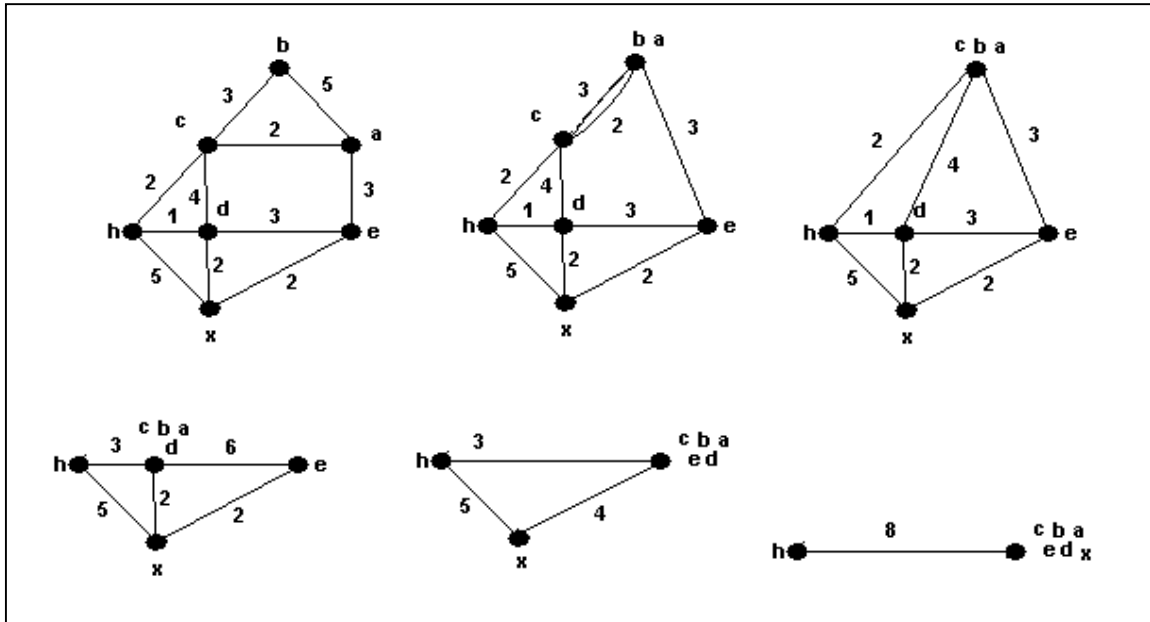


Figura 18 – Segundo passo do algoritmo de minimum-cut

Aqui temos a ordem legal de  $G_{gf} = a, b, c, d, e, x, h$ , onde  $\delta(v_n) = \delta(h) = \{h\}$  e  $u(\delta(h)) = 8$ .

Para o terceiro passo, substituímos  $h$  e  $x$  por  $y$  e  $G_{gf}$  por  $G_{xh}$  e reiniciamos o processo, começando sempre por  $a$ . Teremos a ordem legal de  $G_{xh} = a, b, c, d, e, y$ , onde  $\delta(v_n) = \delta(y) = \{y\}$  e  $u(\delta(y)) = 7$ .

Continua-se a execução até chegar ao passo trivial, na Figura 19.

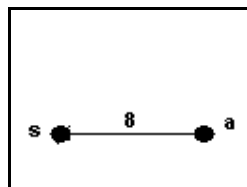


Figura 19 – Último passo (n-1) do algoritmo de *minimum cut*

Temos a ordem legal de  $G_{bt} = a, s$ , onde  $\delta(v_n) = \delta(s) = \{s\}$  e  $u(\delta(s)) = 8$ .

Chegamos ao final do algoritmo com  $M=7$  e  $A=\delta(y)$  (do terceiro passo). Onde  $y = \{x, h\} = \{f, g, h\}$ .

Então  $S = \{f, g, h\}$  é o corte mínimo  $G$  com peso sete.



### 3.3.2 Agrupamento através de *Minimum Spanning Tree*

Uma abordagem prática e interessante em *clustering* para tratamento de informações de genes é apresentada por Tavakoly (Tavakoly,2003) com uso de *minimum spanning tree*(MST).

Cada cluster corresponde a uma sub-árvore da MST e o autor defende que assim, um problema de *clustering* multidimensional é convertido num problema de particionamento de árvore, e enumera suas vantagens:

A estrutura simples de uma árvore facilita implementações eficientes, do contrário seria altamente custoso computacionalmente;

Não depende de uma forma geométrica detalhada do cluster, o que elimina muitas das dificuldades encontradas em algoritmos como o *K-means* e o *Self Organized Map* (SOM) .

#### 3.3.2.1. Algoritmo MST (*Minimum Spanning Tree*)

Em um grafo  $G=(V,E)$  conectado, não direcionado e com pesos, um subconjunto  $T$  de  $E$  cujas arestas criam um caminho entre quaisquer dois nós de  $G$  e a soma dos pesos de suas arestas é mínima, é chamado de *minimum spanning tree*.

Os dois algoritmos clássicos para MST, são *Kruskal* e *Prim*. O algoritmo *Prim* inicia com um vértice qualquer como raiz da árvore e então, num *loop*, seleciona um vértice mais leve (de menor peso) que conecte a árvore corrente a um vértice fora dessa árvore. Ele usa uma fila de prioridades mínimas para todos os vértices que não estão atualmente conectados à árvore que está sendo construída.

O algoritmo de *Kruskal* (Figura 20) cria conjuntos disjuntos para cada vértice de  $V$ . Iterativamente adiciona à árvore que está sendo construída uma aresta de menor peso que conecte dois conjuntos disjuntos. A cada aresta escolhida, os dois vértices se unem num mesmo conjunto. No final, todos os vértices estarão conectados pela árvore gerada.

Seja  $\text{FIND-SET}(u, \mathbb{F})$  um procedimento para selecionar o elemento  $C$  de  $\mathbb{F}$  tal que  $u \in C$ .  
 $G(V, E)$  o grafo de entrada e  $w$  o conjunto de pesos das arestas de  $E$ .

$\text{MST-KRUSKAL}(G, w)$

1.  $A = \{\}, \mathbb{F} = \{\}$
2. Para cada vértice  $v$  em  $V$
3.     Incluir  $\{v\}$  em  $\mathbb{F}$
4. Fim Para
5. Ordene as arestas de  $E$  em ordem decrescente do peso em  $w$
6. Para cada aresta  $(u, v)$  em  $E$ , tomada de  $w$  em ordem decrescente de peso
7.      $C_u = \text{FIND-SET}(u, \mathbb{F}); C_v = \text{FIND-SET}(v, \mathbb{F})$
7.     Se  $C_u \neq C_v$
8.          $A = A \cup \{(u, v)\}$
9.         Remova  $C_u$  de  $\mathbb{F}$
10.        Remova  $C_v$  de  $\mathbb{F}$
11.        Incluir  $C_u \cup C_v$  em  $\mathbb{F}$
12.     Fim Se
13. Fim Para

**retorna**  $A$

**Figura 20 – Algoritmo de Kruskal**

O tempo de execução é  $O(E \log(V))$ , visto que calculado Incluir  $\{v\}$  em  $\mathbb{F}$  leva  $O(V)$  para todos os vértices e ordenar arestas leva  $O(E \log(E))$ , pesquisar cada aresta e encontrar conjunto ( $\text{FIND-SET}$ ) leva  $O(E \alpha(V))$  e união leva  $O(1)$ , o que dá  $O(E \log(E)) = O(E \log(V))$ .

Tavakoly utiliza o algoritmo de *Prim* associado à idéia do *Single-link* dos métodos hierárquicos, que considera a cada passo a união de dois vértices que possuam o menor peso na matriz de dissimilaridade. Com a lista ordenada de arestas da *spanning tree* são retiradas as  $n-1$  arestas de menor peso, são as chamadas arestas inconsistentes. Dessa forma temos  $k$  grupos formados, que representam a solução do problema. O resultado esta na Figura 21 abaixo.

Seja  $D = \{c_1, c_2, \dots, c_n\}$ , o conjunto de elementos.

$A$ , a matriz de adjacência com a distância entre os elementos

$K$ , o número de clusters desejado

$M$ , o mapeamento representado por um conjunto de pares ordenados.

Início

$M = MST(A)$ ;

Identifica arestas inconsistentes em  $M$ ;

Remove  $k - 1$  arestas inconsistentes;

Cria representação de saída;

Fim

**Figura 21 – Algoritmo *Partitional MST***

## Capítulo 4 Estratégia de Solução

A organização dos plantios de eucalipto de uma empresa de fabricação de celulose para garantir madeira de boa qualidade no tempo previsto é uma tarefa complexa e dispendiosa pelo número de variáveis a ser considerado.

Como visto no Capítulo 2 , existem fatores de localização e área de um plantio até a espécie, densidade da madeira e fator de crescimento, idade e outras características próprias dos plantios existentes.

Dentro do grupo de problemas de agrupamento ou *clustering*, devemos escolher uma melhor abordagem que busque uma boa solução para dizer quais serão os grupos ou unidades de corte criados com madeira homogênea ao analista de planejamento.

Das abordagens estudadas e apresentadas anteriormente não poderíamos tirar bom proveito pelo fato dos métodos hierárquicos partirem para uma comparação um a um direta e sem possibilidade de voltar atrás para melhorar o resultado. Já com os métodos de partição temos o problema de indicar um número pré-definido de grupos. A não ser a variação do *k-means* com abordagem de avaliação de limite de similaridade entre elementos do mesmo *cluster* e de *clusters* diferentes, que não precisa da pré-definição, mas ainda tem alta complexidade.

Com isso mais a necessidade de se garantir uma boa qualidade da solução (homogeneidade desses grupos), optamos pela implementação do algoritmo HCS, que garante sobretudo uma forte homogeneidade dos seus grupos.

Para estudarmos o desempenho do algoritmo, vamos utilizar uma simulação de dados reais de plantios de eucalipto.

Neste estudo tínhamos a intenção de comparar o processamento do método HCS com uma abordagem que seguisse os padrões de avaliação de um analista humano. Pela complexidade de se levantar essa metodologia e todas as variáveis consideradas num planejamento real, precisamos simplificar o número de características avaliadas dos plantios e a simulação do analista será feita

por uma abordagem interessante e prática, apesar de ser um algoritmo guloso: agrupamento com uso de *minimum spanning tree*(MST), apresentada no item 3.3.2.

Num primeiro momento vamos avaliar as características de idade (através do ano e mês de plantio), classe de densidade da madeira(1,2,3,4), tipo de regeneração(reforma=0 ou rebrota=1) e localização(latITUDE e longitude).

Nesse contexto vamos avaliar a similaridade (em pontos) da seguinte forma:

- Idade:

**Tabela 1 – Similaridade de idade**

| Diferença de Meses            | Similaridade |
|-------------------------------|--------------|
| diferença < 2                 | 1.0          |
| $2 \leq \text{diferença} < 4$ | 0.6          |
| $4 \leq \text{diferença} < 6$ | 0.2          |
| diferença $\geq 6$            | 0.0          |

- Classe Densidade:

Retorna  $1 - |\text{diferença de classes}| / 3$

Considerando as possíveis classes (1,2,3,4), os resultados serão:

**Tabela 2 – Similaridade da classe de densidade**

| Diferença de Classes | Similaridade |
|----------------------|--------------|
| diferença = 0        | 1.0          |
| diferença = 1        | 2/3          |
| diferença = 2        | 1/3          |
| diferença = 3        | 0.0          |

- Tipo de Regeneração

**Tabela 3 – Similaridade do tipo de regeneração**

| Comparação de tipos | Similaridade |
|---------------------|--------------|
| Tipos iguais        | 1.0          |
| Tipos diferentes    | 0.0          |

- Localização

Retorna  $1 - (|diferença\ de\ latitudes| + |diferença\ de\ longitudes|) / 2$

**Tabela 4 – Similaridade da localização**

| Soma das diferenças de coordenadas | Similaridade |
|------------------------------------|--------------|
| 0.1                                | 0.95         |
| 0.2                                | 0.90         |
| 0.4                                | 0.80         |

Considerando a proximidade em termos de coordenadas, um resultado bom é 0.95, médio é 0.90 e baixo é 0.80. A partir daí é mais baixo ainda.

Avaliando cada item, vamos definir que um plantio é similar a outro se o retorno em cada etapa for respectivamente maior ou igual a 0.6, 0.67, 1.0 e 0.90. Ou seja,

$$\sum_i^n d_i(p_1, p_2) \geq 3.17$$

Nos testes iniciais usamos conjuntos pequenos de plantios para termos uma noção do desenvolvimento do algoritmo.

O primeiro caso foi uma indução direta de dados para gerar o grafo apresentado na discussão sobre HCS (Figura 13) para verificarmos se a implementação geraria os mesmos clusters e gerou. Os outros veremos mais à frente.

#### **4.1. Implementação dos algoritmos**

Apresentamos abaixo a forma de implementação dos algoritmos escolhidos para trabalharem com o problema de agrupamento de áreas florestais.

De modo geral, nosso algoritmo será o da Figura 22.

```
G(V,E) ← GeraGrafoSimilaridade;  
HCS_LOOP(G(V,E));
```

**Figura 22 – Algoritmo de Agrupamento via HCS**

Hartuv e Shamir apenas indicam que a adoção de *singletons* seja feita por um cluster com relativamente próximo ao *singleton* e que este cluster não seja o próprio cluster de *singletons*. Em princípio vamos apenas procurar pelo cluster mais próximo e verificar se ele é diferente do conjunto de *singletons*.

O algoritmo de corte mínimo usado para dar suporte ao HCS é o mesmo da Figura 15, que usa a idéia de identificação de nós.

Nessa abordagem, precisamos de um grafo de similaridade gerado a partir da análise dos dados dos elementos (*fingerprints*), onde basicamente, devemos decidir quais elementos vão estar ligados por arestas devido a sua forte similaridade.

A idéia segue abaixo:

Dados  $c_1, c_2, \dots, c_n$  características de plantio, a similaridade entre dois elementos  $p_1$  e  $p_2$  temos duas opções:

1. Cada  $c_i$  tem uma regra de semelhança, onde  $c_i(p_1) = c_i(p_2)$  ou  $c_i(p_1) \neq c_i(p_2)$
2. A regra dá uma pontuação (grau de semelhança) entre  $c_i(p_1)$  e  $c_i(p_2)$  .

De qualquer forma deve-se definir um marco inicial (*threshold*)  $t$  na opção 1 para tomar a decisão e na opção 2 para criação de uma aresta entre  $p_1$  e  $p_2$  caso

$$\sum_i^n \text{grauDeSemelhanca}(c_i(p_1), c_i(p_2)) > t. \quad (8)$$

Optando-se então, por 2, criamos a aresta se  $\sum_i^n \text{grauDeSemelhanca}(c_i(p_1), c_i(p_2)) > t$ , já que cada característica possui uma forma de avaliação diferente, muitas vezes não permitindo o uso de uma fórmula de cálculo de similaridade única.

Assim temos o algoritmo da Figura 23 para geração de grafo de similaridade:

```
GeraGrafoSimilaridade(conjunto de  $n$  elementos  $P$ )
Crie um grafo  $G$  com  $n$  vértices.
Para  $i=0$  até  $n-1$  faça
    Para  $j=i+1$  até  $n-1$  faça
        Se  $\sum_i^n \text{grauDeSemelhanca}(c_i(p_1), c_i(p_2)) > m$  então
            Crie uma aresta  $(i,j)$ .
        Fim Para
    Fim Para
Fim Para
Retorne  $G$ 
```

**Figura 23 – Algoritmo *GeraGrafoSimilaridade***

Este algoritmo tem complexidade da ordem  $O(n^2)$ .

Para efetuar a comparação com outro método, tomamos a implementação da abordagem de Tavakoly aplicando o método de *Kruskal*.

Um detalhe importante desta implementação é o fato de trabalharmos com valores de similaridade. Neste caso deveríamos pensar na idéia de *Complete link* e não *Single link*, da avaliação de similaridade dos métodos hierárquicos, que busca ligações de maior peso. Entretanto, teríamos um problema com a solução do MST, pois aí já seria interessante mantermos somente as arestas de maior peso. A alternativa então, é modificar o algoritmo de MST para buscar a máxima *spanning tree* e a idéia de *Complete link*. No final retiramos as arestas de menor peso.

Outra questão é quanto à necessidade de informar o número de *clusters* desejado. O que fazer se não queremos informar  $k$ ?

Uma solução é usar o *threshold*  $m$  calculado para o HCS de forma que as arestas com peso menor que  $m$  sejam retiradas, formando os *clusters*.



Assim, temos o algoritmo da Figura 24 para agrupamento:

|        |  |
|--------|--|
| Sejam  | <p>FIND-SET(<math>u, S</math>) um procedimento para selecionar o elemento <math>C</math> de <math>S</math> tal que <math>u \in C</math>.</p> <p><math>G(V, E)</math> o grafo de entrada e <math>w</math> o conjunto de pesos das arestas de <math>E</math>.</p> <p>MST-KRUSKAL-INVERTIDO(<math>G, w</math>) a função MST-KRUSKAL ordenando arestas de maior peso.</p> <p>MAKE-SET(<math>v, S</math>) : incluir <math>\{v\}</math> em <math>S</math></p> <p>UNION(<math>u, v, S</math>) :    Remova <math>C_u</math> de <math>S</math><br/>                                     Remova <math>C_v</math> de <math>S</math><br/>                                     Incluir <math>C_u \cup C_v</math> em <math>S</math></p> <p><math>t</math> um <i>threshold</i> para a avaliação de similaridade</p> |
| Início | <ol style="list-style-type: none"> <li>1. <math>S = \{\}</math></li> <li>2. <math>A = \text{MST-KRUSKAL-INVERTIDO}(G, w)</math></li> <li>3. Para cada vértice <math>v</math> em <math>V</math></li> <li>4.     <math>S = S \cup \text{MAKE-SET}(v, S)</math></li> <li>5. Fim Para</li> <li>6. Para cada aresta(<math>u, v</math>) em <math>A</math></li> <li>7.     <math>C_u = \text{FIND-SET}(u, S)</math></li> <li>8.     <math>C_v = \text{FIND-SET}(v, S)</math></li> <li>9.     Se <math>C_u \neq C_v</math> e <math>w(u, v) &gt; t</math></li> <li>10.        UNION(<math>u, v, S</math>)</li> <li>11.     Fim Se</li> <li>12. Fim Para</li> <li>13. Retorna <math>S</math></li> </ol>  |

**Figura 24 – Algoritmo MST-Clustering Invertido**

Aqui o grafo  $G(V, E)$  é um grafo completo, na verdade um clique, pois cada vértice tem uma ligação com todos os outros. Para o tratamento do HCS temos um grafo apenas com as arestas que ultrapassam o *threshold* e sem peso. Dessa forma vamos usar uma estrutura auxiliar para guardar a matriz de adjacências, com a similaridade (peso) de cada aresta para o MST.

De qualquer forma, o processamento para construção do grafo de similaridade e da matriz de adjacências é o mesmo: comparar cada elemento com os outros  $n-1$  calculando seu grau de similaridade. A diferença é que a decisão tomada em comparação com o *threshold* acontece antes do processamento para o algoritmo HCS e no final do MST-Clustering invertido.

## 4.2. Comparação dos métodos

Usando a informação da seção anterior sobre a avaliação de similaridade, executamos alguns pequenos testes para verificar e entender o comportamento dos algoritmos.

A entrada para os programas é um arquivo texto com um conjunto de plantios em que as colunas representam identificador do plantio, grupo, ano de plantio, mês de plantio, classe de densidade, tipo de regeneração, latitude e longitude.

No primeiro conjunto (Figura 25) temos:

|    |   |      |    |   |          |        |
|----|---|------|----|---|----------|--------|
| 0  | 0 | 1999 | 2  | 4 | 0 -19.20 | -41.00 |
| 1  | 0 | 1999 | 2  | 2 | 0 -19.30 | -40.50 |
| 2  | 0 | 2000 | 1  | 1 | 0 -19.30 | -40.00 |
| 3  | 0 | 2000 | 1  | 3 | 0 -19.50 | -40.50 |
| 4  | 0 | 2000 | 2  | 1 | 0 -19.50 | -40.10 |
| 5  | 0 | 1999 | 12 | 1 | 0 -19.50 | -40.00 |
| 6  | 0 | 1999 | 12 | 2 | 0 -19.20 | -40.10 |
| 7  | 0 | 1998 | 9  | 1 | 0 -19.10 | -40.10 |
| 8  | 0 | 1998 | 9  | 3 | 0 -19.20 | -40.50 |
| 9  | 0 | 1998 | 9  | 1 | 0 -19.10 | -40.50 |
| 10 | 0 | 1999 | 2  | 2 | 0 -19.10 | -41.00 |
| 11 | 0 | 1999 | 2  | 1 | 0 -19.20 | -40.00 |

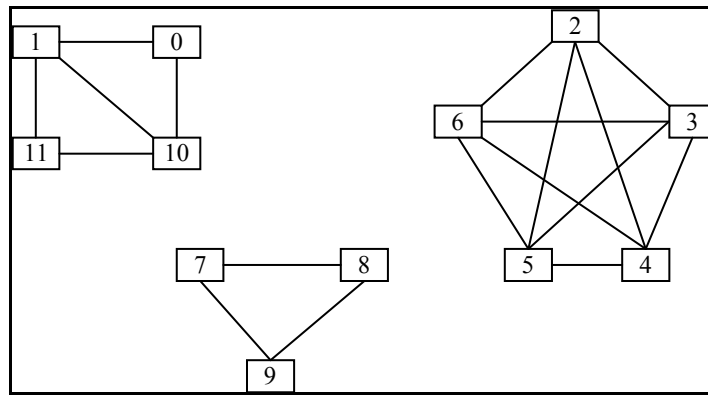
Figura 25 – Conjunto de plantios 1

O conjunto gera a matriz de adjacências da Figura 26 abaixo:

|      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
|      | [0]  | [1]  | [2]  | [3]  | [4]  | [5]  | [6]  | [7]  | [8]  | [9]  | [10] | [11] |
| [0]  | 0    |      |      |      |      |      |      |      |      |      |      |      |
| [1]  | 3,7  | 0    |      |      |      |      |      |      |      |      |      |      |
| [2]  | 1,45 | 2,75 | 0    |      |      |      |      |      |      |      |      |      |
| [3]  | 2,6  | 2,9  | 3,65 | 0    |      |      |      |      |      |      |      |      |
| [4]  | 1,4  | 2,7  | 3,85 | 3,8  | 0    |      |      |      |      |      |      |      |
| [5]  | 1,35 | 2,65 | 3,9  | 3,75 | 3,55 | 0    |      |      |      |      |      |      |
| [6]  | 2,55 | 2,75 | 3,9  | 3,65 | 3,45 | 3,8  | 0    |      |      |      |      |      |
| [7]  | 1,7  | 2,9  | 2,85 | 2,6  | 2,8  | 2,75 | 2,95 | 0    |      |      |      |      |
| [8]  | 2,95 | 3,15 | 2,7  | 2,85 | 2,65 | 2,6  | 2,8  | 3,75 | 0    |      |      |      |
| [9]  | 1,9  | 3,1  | 2,65 | 2,8  | 2,6  | 2,55 | 2,75 | 3,8  | 3,95 | 0    |      |      |
| [10] | 3,95 | 3,65 | 2,4  | 2,55 | 2,35 | 2,3  | 2,5  | 2,75 | 2,9  | 2,95 | 0    |      |
| [11] | 2,5  | 3,7  | 2,95 | 2,6  | 2,8  | 2,85 | 2,95 | 3,1  | 2,95 | 2,9  | 3,45 | 0    |
|      | [0]  | [1]  | [2]  | [3]  | [4]  | [5]  | [6]  | [7]  | [8]  | [9]  | [10] | [11] |

Figura 26 – Matriz de Adjacências 1

Para o HCS, o grafo de similaridade gerado (Figura 27) foi:



**Figura 27 – Grafo de Similaridade 1**

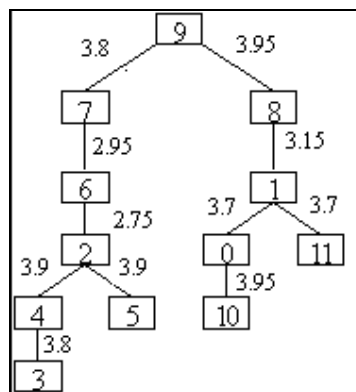
E na Figura 28 temos os grupos homogêneos (*clusters*) gerados.

|    |   |      |    |   |   |       |       |
|----|---|------|----|---|---|-------|-------|
| 7  | 0 | 1998 | 9  | 1 | 0 | -19.1 | -40.1 |
| 8  | 0 | 1998 | 9  | 3 | 0 | -19.2 | -40.5 |
| 9  | 0 | 1998 | 9  | 1 | 0 | -19.1 | -40.5 |
| 2  | 1 | 2000 | 1  | 1 | 0 | -19.3 | -40.0 |
| 5  | 1 | 1999 | 12 | 1 | 0 | -19.5 | -40.0 |
| 6  | 1 | 1999 | 12 | 2 | 0 | -19.2 | -40.1 |
| 3  | 1 | 2000 | 1  | 3 | 0 | -19.5 | -40.5 |
| 4  | 1 | 2000 | 2  | 1 | 0 | -19.5 | -40.1 |
| 0  | 2 | 1999 | 2  | 4 | 0 | -19.2 | -41.0 |
| 1  | 2 | 1999 | 2  | 2 | 0 | -19.3 | -40.5 |
| 10 | 2 | 1999 | 2  | 2 | 0 | -19.1 | -41.0 |
| 11 | 2 | 1999 | 2  | 1 | 0 | -19.2 | -40.0 |

**Figura 28 – Clusters por HCS 1**

Um resultado muito esperado já que o grafo é composto de três subgrafos desconexos entre si.

Para o MST-Clustering invertido, a *spanning tree* gerada a da Figura 29 abaixo:



**Figura 29 – Máxima Spanning Tree 1**

E o máximo *spanning tree* gerou o mesmo conjunto de clusters, dada a desconexão entre eles.

Veja que as únicas arestas com peso menor que o *threshold*  $t = 3.17$ , definido no início deste capítulo, são (7,6) e (8,1). Ao retirá-las da árvore, ficamos com os três clusters.

O segundo exemplo (Figura 30) foi:

|    |   |      |   |   |   |        |        |
|----|---|------|---|---|---|--------|--------|
| 0  | 0 | 1999 | 2 | 1 | 0 | -19.20 | -41.00 |
| 1  | 0 | 1999 | 2 | 3 | 1 | -19.10 | -40.90 |
| 2  | 0 | 1999 | 2 | 2 | 1 | -19.30 | -41.90 |
| 3  | 0 | 1999 | 6 | 4 | 0 | -19.10 | -40.70 |
| 4  | 0 | 1999 | 4 | 1 | 0 | -19.30 | -40.70 |
| 5  | 0 | 1999 | 8 | 4 | 0 | -19.10 | -40.50 |
| 6  | 0 | 1999 | 4 | 2 | 0 | -19.30 | -40.50 |
| 7  | 0 | 1999 | 8 | 1 | 0 | -19.10 | -40.40 |
| 8  | 0 | 1999 | 8 | 1 | 0 | -19.10 | -40.30 |
| 9  | 0 | 1999 | 6 | 1 | 0 | -19.30 | -40.30 |
| 10 | 0 | 1999 | 6 | 2 | 0 | -19.10 | -41.30 |
| 11 | 0 | 1999 | 8 | 1 | 0 | -19.20 | -40.50 |

**Figura 30 – Conjunto de plantios 2**

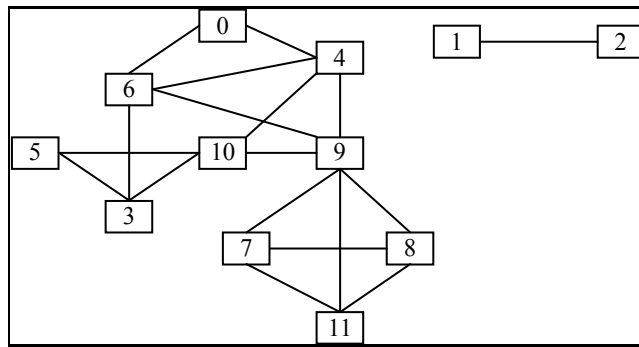
Com a seguinte (Figura 31) matriz de adjacências:

|      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
|      | [0]  | [1]  | [2]  | [3]  | [4]  | [5]  | [6]  | [7]  | [8]  | [9]  | [10] | [11] |
| [0]  | 0    |      |      |      |      |      |      |      |      |      |      |      |
| [1]  | 2,9  | 0    |      |      |      |      |      |      |      |      |      |      |
| [2]  | 2,5  | 3,4  | 0    |      |      |      |      |      |      |      |      |      |
| [3]  | 2    | 2,1  | 1,5  | 0    |      |      |      |      |      |      |      |      |
| [4]  | 3,4  | 2,4  | 2    | 2,5  | 0    |      |      |      |      |      |      |      |
| [5]  | 1,7  | 1,8  | 1,2  | 3,5  | 2    | 0    |      |      |      |      |      |      |
| [6]  | 3,3  | 2,3  | 1,9  | 3,4  | 3,9  | 3,1  | 0    |      |      |      |      |      |
| [7]  | 2,65 | 1,75 | 1,15 | 2,45 | 2,95 | 2,95 | 3,05 | 0    |      |      |      |      |
| [8]  | 2,6  | 1,7  | 1,1  | 2,4  | 2,9  | 2,9  | 3    | 3,95 | 0    |      |      |      |
| [9]  | 2,8  | 1,8  | 1,4  | 2,7  | 3,4  | 2,4  | 3,5  | 3,45 | 3,5  | 0    |      |      |
| [10] | 3    | 2    | 1,8  | 3,7  | 3,2  | 3,2  | 3,1  | 3,15 | 3,1  | 3,4  | 0    |      |
| [11] | 2,75 | 1,75 | 1,25 | 2,45 | 3,05 | 2,95 | 3,15 | 3,9  | 3,85 | 3,45 | 3,15 | 0    |
|      | [0]  | [1]  | [2]  | [3]  | [4]  | [5]  | [6]  | [7]  | [8]  | [9]  | [10] | [11] |

**Figura 31 – Matriz de Adjacências 2**

Aqui temos uma diferença menor de idade, mas foi introduzida uma indicação de tipo de regeneração diferente em dois plantios.

O grafo de similaridade gerado (Figura 32) para o HCS foi:



**Figura 32 – Grafo de Similaridade 2**

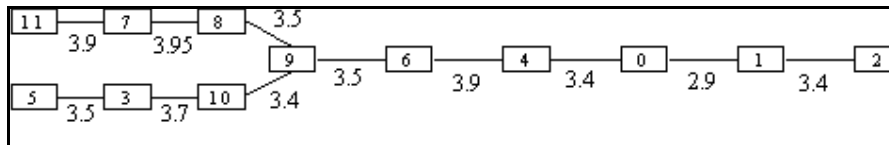
E o resultado (Figura 33) foi:

|    |   |      |   |   |   |       |       |
|----|---|------|---|---|---|-------|-------|
| 10 | 0 | 1999 | 6 | 2 | 0 | -19.1 | -41.3 |
| 3  | 0 | 1999 | 6 | 4 | 0 | -19.1 | -40.7 |
| 5  | 0 | 1999 | 8 | 4 | 0 | -19.1 | -40.5 |
| 4  | 1 | 1999 | 4 | 1 | 0 | -19.3 | -40.7 |
| 6  | 1 | 1999 | 4 | 2 | 0 | -19.3 | -40.5 |
| 0  | 1 | 1999 | 2 | 1 | 0 | -19.2 | -41.0 |
| 7  | 2 | 1999 | 8 | 1 | 0 | -19.1 | -40.4 |
| 8  | 2 | 1999 | 8 | 1 | 0 | -19.1 | -40.3 |
| 9  | 2 | 1999 | 6 | 1 | 0 | -19.3 | -40.3 |
| 11 | 2 | 1999 | 8 | 1 | 0 | -19.2 | -40.5 |
| 1  | 3 | 1999 | 2 | 3 | 1 | -19.1 | -40.9 |
| 2  | 3 | 1999 | 2 | 2 | 1 | -19.3 | -41.9 |

**Figura 33 – Clusters por HCS 2**

Os dois plantios com tipo de regeneração diferente ficaram num grupo à parte.

E o máximo *spanning tree* gerou apenas dois *clusters*, dada a *spanning tree* da Figura 34 abaixo:



**Figura 34 – Máxima SpanningTree 2**

Veja que somente a aresta (1,0) está abaixo de 3.17. Diferente do HCS, aqui não se separou (6,9) e (9,10), cujos custos são 3.5 e 3.4 respectivamente. Considerando que o HCS trabalhou com um grafo em que só havia arestas com peso superior a 3.17, os seus clusters são mais homogêneos pela definição de que são altamente conectados ao terem, cada um, um *minimum cut* com número igual

ou superior à metade dos seus vértices. Observe também que, a união de 3,5 e 10 ao conjunto {7,8,9,11} se faz completamente por arestas de peso menor que 3.17, exceto (10,9), o que nos leva a refletir que uma análise mais criteriosa realmente subdividiria o grupo.

### **4.3. Novos critérios**

Num caso real o tipo de regeneração não seria muito útil para separação de corte já que não influencia diretamente na produção. As classes de densidade, que no real seriam fortes separadores, de forma que plantios de classes diferentes não poderiam ficar juntos, poderiam ser substituídas por densidade simplesmente, representada por um número inteiro. A similaridade entre densidades pode ser calculada então, de forma contínua, tomando por base o mínimo e máximo conhecidos e sua diferença como pior caso. Densidades iguais são o melhor caso. Por último, seria interessante usar o fator de produtividade (boa, média, ruim) de um plantio, de forma que se a produtividade for boa aumentamos a idade em 6 meses, pois podemos colher mais cedo e se a produtividade for ruim, aumentamos em 1 ano, pois é melhor começar novo plantio do que manter um de alto custo. Aproveitando, ano e mês de plantio podem ser substituídos por idade simplesmente, representada por um número decimal para dar a precisão dos meses.

Nesse contexto vamos avaliar a similaridade (em pontos) da seguinte forma:

- Idade:

Primeiro avalia-se a produtividade, somando-se um fator à idade como um ajuste para a colheita.

**Tabela 5 – Ajuste de Produtividade na Idade**

| <b>Produtividade</b> | <b>Idade</b> |
|----------------------|--------------|
| B (boa)              | + 0.5        |
| M (média)            | ----         |
| R (ruim)             | + 1.0        |

**Tabela 6 – Similaridade de idade**

| Diferença de Meses | Similaridade |
|--------------------|--------------|
| diferença < 2      | 1.0          |
| 2 ≤ diferença < 4  | 0.6          |
| 4 ≤ diferença < 6  | 0.2          |
| diferença ≥ 6      | 0.0          |

- Classe Densidade:

Retorna  $1 - |diferença\ de\ densidade| / 240$

Como as classes variam de (417...645), pegamos a diferença e dividimos por 240 para fazer uma normalização e termos um valor entre 0 e 1.

- Localização

Retorna  $1 - \left( \frac{\sqrt{|diferença\ de\ latitudes|^2 + |diferença\ de\ longitudes|^2}}{\sqrt{2}} \right)$

Aqui também dividimos por um valor normalizador para ter uma similaridade entre 0 e 1. O valor  $\sqrt{2}$  vem do fato de se considerar 1 grau de latitude e 1 grau de longitude um limite razoável para manter áreas num mesmo grupo, não sendo viável processar o algoritmo para localizações mais distantes.

Assim, para 0.1 de latitude e longitude, temos um valor bom = 0.9.

Avaliando cada item, vamos definir que um plantio é similar a outro se o retorno em cada etapa for respectivamente maior ou igual a 0.6, 0.5 e 0.90. Ou seja,

$$\sum_i^n s_i(p_1, p_2) \geq 2$$

### 4.3.1 Novo exemplo

Em mais um exemplo para esse novo formato de avaliação temos o arquivo de entrada com conjuntos de plantios (Figura 35) onde as colunas representam identificador do plantio, grupo, idade, densidade, produtividade, latitude e longitude.

|   |   |     |     |   |           |           |
|---|---|-----|-----|---|-----------|-----------|
| 0 | 0 | 5,3 | 527 | B | 18,593333 | 39,732222 |
| 1 | 0 | 5,1 | 527 | B | 18,599444 | 39,714444 |
| 2 | 0 | 4,8 | 561 | B | 18,573333 | 39,698333 |
| 3 | 0 | 5,3 | 455 | B | 18,573611 | 39,680833 |
| 4 | 0 | 4,8 | 561 | B | 18,557222 | 39,686667 |
| 5 | 0 | 5,3 | 455 | B | 18,558889 | 39,702778 |
| 6 | 0 | 5,3 | 455 | B | 18,548333 | 39,722222 |
| 7 | 0 | 4,8 | 527 | B | 18,539167 | 39,737500 |
| 8 | 0 | 4,8 | 575 | B | 18,562500 | 39,763056 |
| 9 | 0 | 4,8 | 575 | B | 18,573889 | 39,748889 |

Figura 35 – Conjunto de Plantios 3

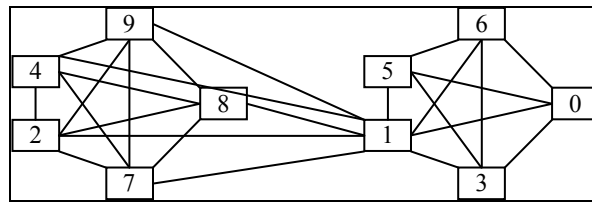
A matriz de similaridade está na Figura 36:

|     |      |      |      |      |      |      |      |      |      |     |
|-----|------|------|------|------|------|------|------|------|------|-----|
|     | [0]  | [1]  | [2]  | [3]  | [4]  | [5]  | [6]  | [7]  | [8]  | [9] |
| [0] | 0    |      |      |      |      |      |      |      |      |     |
| [1] | 2,18 | 0    |      |      |      |      |      |      |      |     |
| [2] | 1,97 | 2,17 | 0    |      |      |      |      |      |      |     |
| [3] | 2,15 | 2,16 | 1,99 | 0    |      |      |      |      |      |     |
| [4] | 1,95 | 2,16 | 2,18 | 1,99 | 0    |      |      |      |      |     |
| [5] | 2,16 | 2,16 | 1,99 | 2,18 | 1,99 | 0    |      |      |      |     |
| [6] | 2,16 | 2,16 | 1,97 | 2,16 | 1,97 | 2,18 | 0    |      |      |     |
| [7] | 1,95 | 2,15 | 2,16 | 1,94 | 2,15 | 1,97 | 1,99 | 0    |      |     |
| [8] | 1,96 | 2,15 | 2,14 | 1,93 | 2,14 | 1,95 | 1,96 | 2,17 | 0    |     |
| [9] | 1,98 | 2,16 | 2,16 | 1,94 | 2,15 | 1,96 | 1,97 | 2,17 | 2,18 | 0   |
|     | [0]  | [1]  | [2]  | [3]  | [4]  | [5]  | [6]  | [7]  | [8]  | [9] |

Figura 36 – Matriz de Adjacências 3

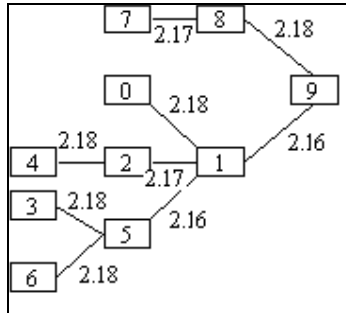
O grafo de similaridade é representado na Figura 37:





**Figura 37 – Grafo de Similaridade 3**

E a máxima *spanning tree*, na Figura 38:



**Figura 38 – Máxima *Spanning Tree* 3**

Como nenhuma das arestas tem peso menor que 2, o resultado é o próprio grafo. Novamente, se analisarmos a similaridade de 0, 3, 5 e 6 com os elementos do conjunto {1,2,4,7,8,9} podemos perceber que só há ligação forte (peso>2) com o vértice 1. Portanto, uma análise mais criteriosa subdividiria o grafo.

#### **4.4. Melhorias importantes**

A procura por cortes mínimos muitas vezes leva ao surgimento de *singletons*. Para evitar isso e minimizar o processo de adoções, optamos por não aceitar cortes que causem *singletons*. Para o primeiro exemplo, tudo bem. Entretanto, numa primeira execução do exemplo 2 notamos que o vértice 10 havia ficado com o grupo do vértice 9 e isso ocorreu pelo fato do *minimum cut* achar dois valores mínimos intermediários iguais -  $\delta(v_n)$  (apresentado no item 3.3.1.1) é igual mas,  $v_n$  de um passo é maior que  $v_n$  de outro passo. Como só trocávamos de *cut* com um valor menor, ficávamos com o primeiro *cut* encontrado, que gerava um subgrafo com menos vértices. Assim, passamos a escolher o *cut* que gere um subgrafo com mais vértices em caso de igualdade de valor.

Sobre a adoção de *singletons*, será que esta adoção pode estar acabando com a homogeneidade do cluster recém gerado que os está adotando?

Na dúvida podemos reprocessar o cluster. Mas, assim podemos causar um *loop* infinito no processo.

Então como identificar se o *cluster* tem um número suficientemente grande de vizinhos para executar adoção como fala o algoritmo em 3.3.1?

Para isso optamos por adotar o *singleton* somente se ele tiver uma proximidade com o cluster maior ou igual ao grau de conectividade do cluster ( $k(G)$ ). Assim, temos a certeza de que o cluster continuará altamente conectado.

## **4.5. Modelo de classes**

No projeto montado para a implementação dos algoritmos foram definidas, inicialmente classes para atender às necessidades do algoritmo HCS, depois foram acrescentadas outras classes para atender ao algoritmo MST-*Clustering* invertido.

### **4.5.1 Classes para o HCS**

A classe Elemento contém toda a informação necessária sobre os plantios com os métodos para leitura e escrita dessa informação. Veja a Figura 39.

A classe vértice contém um Elemento ou plantio mais uma lista de vértices adjacentes e uma lista de componentes. Componentes são necessários para implementar a união de vértices no processamento do HCS, onde o algoritmo de Ordem Legal une dois vértices, gerando um novo, mas precisamos recuperar os vértices que formaram esse novo no final para termos o conjunto separado pelo *minimum cut*. Por esse motivo também são necessários outros métodos para verificar se o vértice tem ou não componentes e se está vazio, além dos acessos naturais de leitura e escrita das listas.

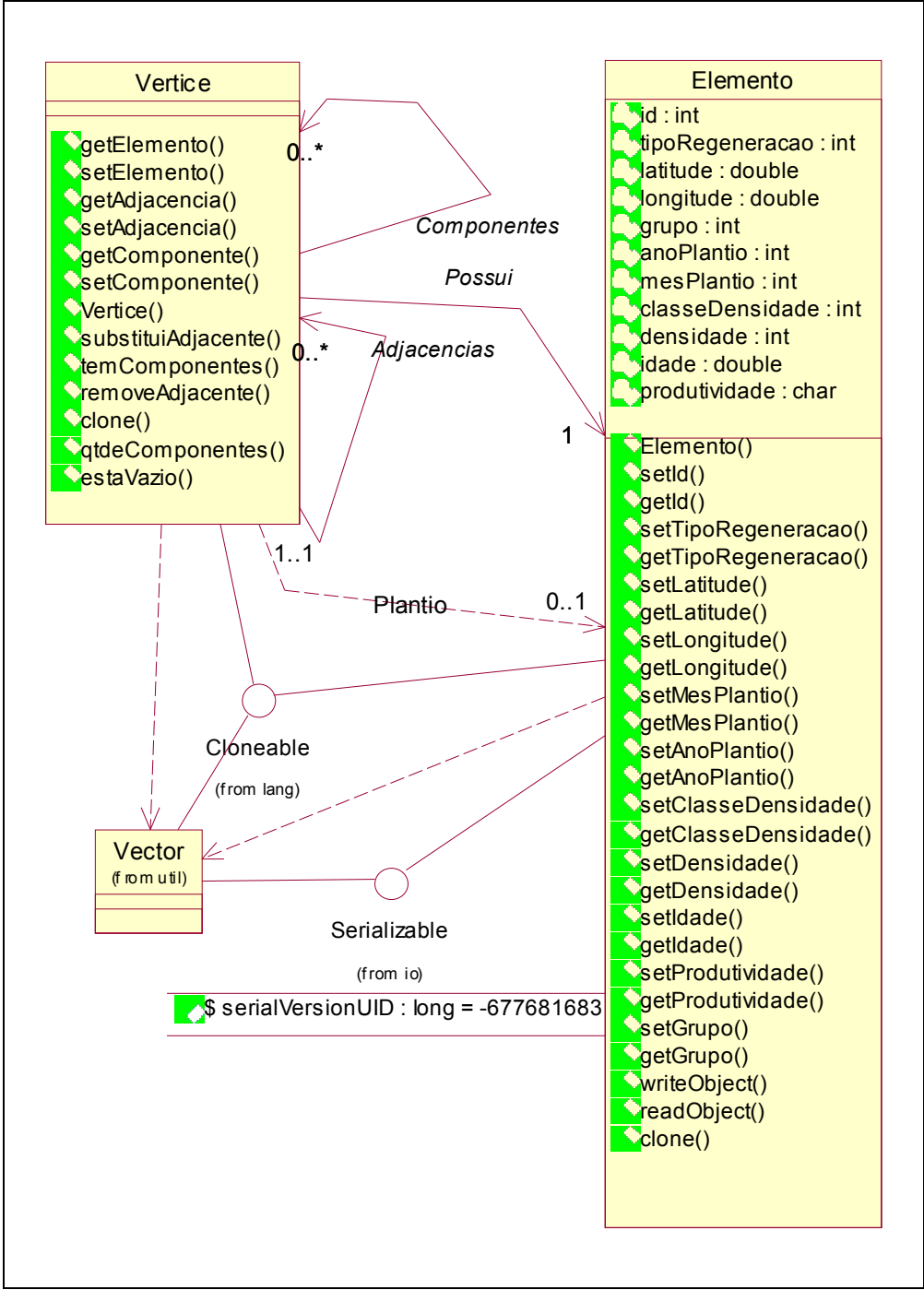


Figura 39 – Modelo de Classes para HCS (Elemento e Vértice)

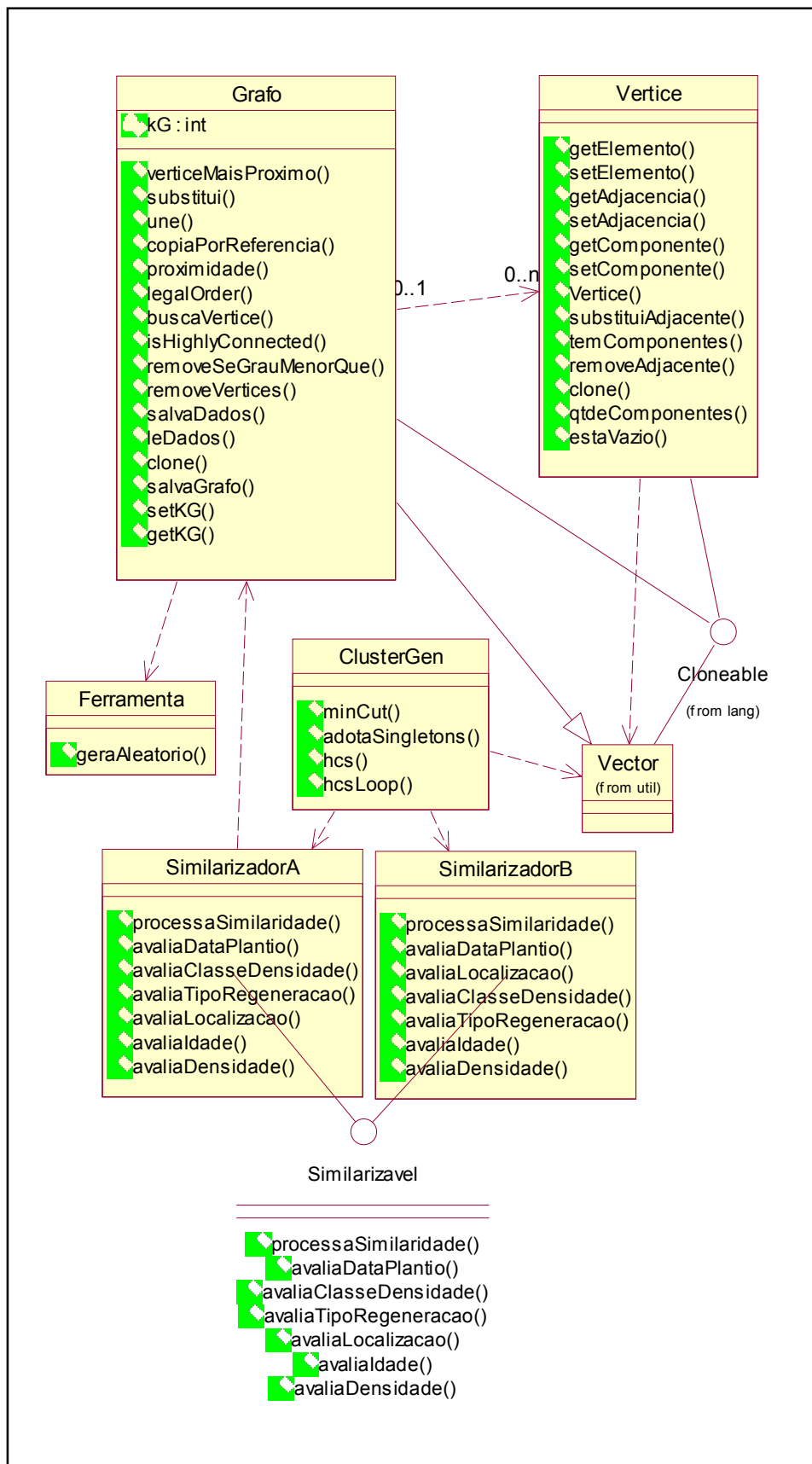


Figura 40 – Modelo de Classes para HCS

Na Figura 40 temos as outras classes necessárias ao HCS.

Grafo é um conjunto de vértices que guarda a informação adicional da conectividade de  $G$ ,  $k(G)$ . A lista de adjacências está com o vértice. Além disso, possui métodos para buscar, inserir e remover vértices, métodos para ler dados de um arquivo texto, criar um grafo e gravar o resultado do grafo de similaridade mais os *clusters* gerados de volta em arquivos texto e métodos para unir vértices e gerar clone do grafo. Ainda possui alguns métodos bem típicos do algoritmo:

- *legalOrder* para auxiliar o *minimum cut* com a ordem legal dos vértices do grafo.
- *isHighlyConnect* - indica se o grafo é altamente conexo, ou seja, se tem  $n > 1$  vértices e  $k(G) > n/2$ . Onde  $k(G)$  é o mínimo de arestas cuja remoção resulta num grafo desconexo.
- *proximidade* - Calcula o número de arestas que unem o grafo ao vértice passado como parâmetro.
- *verticeMaisProximo* - Para cada vértice do grafo, percorre todos os vértices de um grafo 2 passado como parâmetro, analisando suas listas de adjacências para saber qual o vértice que tem mais adjacentes em  $g2$ .
- *removeSeGrauMenorQue* - Remove os vértices com grau menor que um mínimo  $d$  e retorna o subgrafo de vértices retirados.

ClusterGen é a classe que implementa o algoritmo HCS com seus métodos *hcs*, *hcsloop*, *mincut* e *adotaSingletons*. Todos refletindo o planejamento da estratégia de solução.

*Similarizavel* é uma interface que indica os processos de cálculo da similaridade entre vértices. É composta dos métodos de avaliação de similaridade para cada característica de plantio definida anteriormente e o método *processaSimilaridade* que dá o resultado final da similaridade entre dois vértices.

Temos duas implementações para a interface: *SimilarizadorA* e *SimilarizadorB*. Isso se deve ao fato de termos criado uma nova avaliação de similaridade no decorrer do estudo. *SimilarizadorA* processa a similaridade considerando tipo de regeneração e as outras características mostradas

inicialmente, enquanto que *SimilarizadorB* calcula a similaridade avaliando produtividade, densidade e a nova normalização da segunda alternativa de critérios mostrada no item 4.3.

*Ferramenta* é uma classe auxiliar para métodos de uso geral como gerar números aleatórios.

*Vector* e *Cloneable* são especificidades do projeto apenas para representar a forma de armazenamento das adjacências dos vértices e da possibilidade de criar-se clone deles para o processamento.

#### **4.5.2 Classes para o MST-Clustering invertido**

A mesma estrutura para *Elemento* é usada na implementação do algoritmo baseado no MST.

Também são usadas as classes *Vértice* e *Grafo* como armazenamento dos dados dos plantios.

*MatrizAdj* é a classe para a matriz de adjacências do grafo dos plantios. Contém todos os custos das arestas entre os vértices do conjunto de plantios da entrada de dados. Foi criada posteriormente porque o grafo de similaridade do HCS era construído diretamente da leitura dos dados dos plantios e após a avaliação de similaridade, arestas eram criadas sem peso ou custo.

*MstSimilarizavel* tem o método para gerar a matriz de adjacências pela análise da similaridade.

*MstSimilarizadorA* e *MstSimilarizadorB* – geram a matriz de adjacências segundo os dois diferentes grupos de critérios e formas de avaliação comentados anteriormente.

*MstClusterGen* implementa o MST-Clustering invertido com um método principal *geraClusters* para controlar a impressão dos resultados e a chamada do algoritmo de Kruskal, pelo método de mesmo nome, que por sua vez, usa os métodos auxiliares *makeSet*, *union* e *findSet* como descritos no item 4.1. O outro método auxiliar é *ordenaPorCusto*, que retorna as arestas ordenadas de forma decrescente por custo.

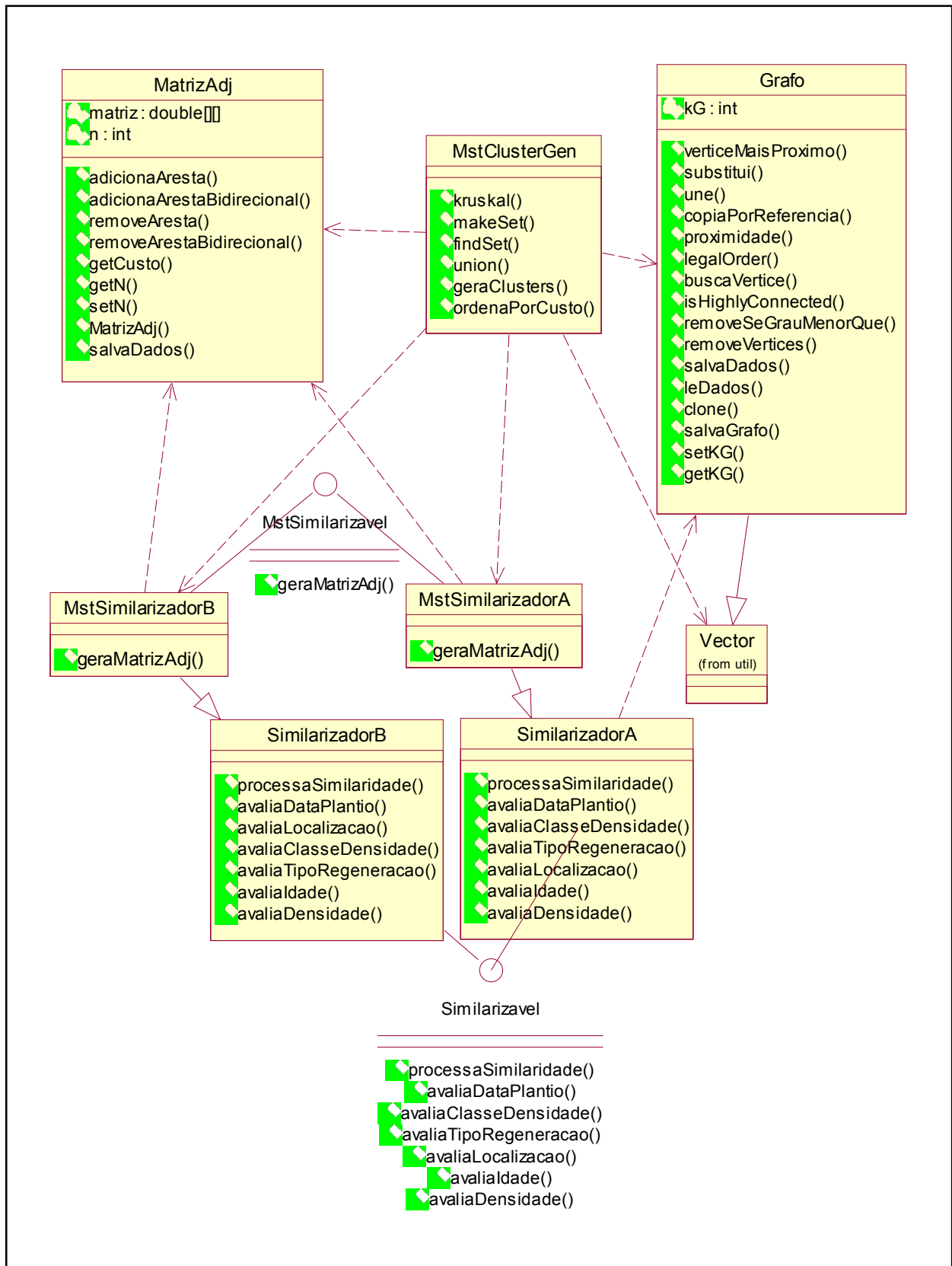


Figura 41 – Modelo de Classes para MST-Clustering invertido

## Capítulo 5 Testes Numéricos

Os testes foram realizados em um computador Pentium II 400 MHz através do Oracle Jdeveloper em Windows 2000.

O padrão de entrada de dados e avaliação de similaridade usados foi o segundo, após a mudança de critérios explicada na seção 4.3 e tratado pelo *SimilarizadorB* e *MstSimilarizadorB* das classes apresentadas no final da seção 4.5.

Um exemplo inicial é um conjunto de 50 plantios mostrado na Figura 42. Sua matriz de adjacências é apresentada na Tabela 7.

O HCS gerou os grupos {35,39,41,38,43,45}, {1,2,4,7,8,9,12,13,15,17,19,20,21,22,23,24,25,27,29,31,33,34,36,47,48}, {30,40,46,32,37}, {0,3,5,6,14,16,18,26,28,49} e {10,11,42,44}.

Os grupos gerados pelo MST-Ci foram {10,11}, {38,41,35,39,43,45}, {42,44} e outro com o restante.

No grupo maior podemos perceber ligações fracas como nos exemplos do Capítulo 4 , já a separação entre {10,11} e {42,44} faz sentido, já que as ligações entre os grupos são fracas.No HCS, excluimos do processamento vértices de grau baixo (no caso  $< 2$ ) e criamos um grupo deles no final, por isso o grupo com os quatro vértices.

Ainda executamos para dois grupos de 100 plantios e um de 484 plantios.

No primeiro grupo de 100, apenas acrescentamos 50 plantios ao grupo anterior. Os algoritmos geraram a mesma quantidade de grupos, inclusive os elementos dos grupos anteriores permaneceram nos seus conjuntos ({10,11}{42,44} ou {10,11,42,44}) e receberam novos componentes como {38,41,35,39,43,45,59,54,55,52,53}.

Para o segundo caso de 100 elementos, selecionamos plantios com dois grupos de localização nitidamente diferente e em um deles uma diferença também nítida em termos de idade. O resultado foi o mesmo para os dois algoritmos, que geraram 3 grupos de características semelhantes.



|    |   |      |       |           |           |
|----|---|------|-------|-----------|-----------|
| 0  | 0 | 5,30 | 527 B | 18,593333 | 39,732222 |
| 1  | 0 | 5,10 | 527 B | 18,599444 | 39,714444 |
| 2  | 0 | 4,80 | 561 B | 18,573333 | 39,698333 |
| 3  | 0 | 5,30 | 455 B | 18,573611 | 39,680833 |
| 4  | 0 | 4,80 | 561 B | 18,557222 | 39,686667 |
| 5  | 0 | 5,30 | 455 B | 18,558889 | 39,702778 |
| 6  | 0 | 5,30 | 455 B | 18,548333 | 39,722222 |
| 7  | 0 | 4,80 | 527 B | 18,539167 | 39,737500 |
| 8  | 0 | 4,80 | 575 B | 18,562500 | 39,763056 |
| 9  | 0 | 4,80 | 575 B | 18,573889 | 39,748889 |
| 10 | 0 | 9,00 | 509 B | 18,593333 | 39,714167 |
| 11 | 0 | 9,00 | 504 B | 18,614444 | 39,690278 |
| 12 | 0 | 4,80 | 561 B | 18,606944 | 39,703889 |
| 13 | 0 | 4,70 | 561 B | 18,599444 | 39,753611 |
| 14 | 0 | 5,20 | 561 B | 18,549444 | 39,670833 |
| 15 | 0 | 4,60 | 509 B | 18,532778 | 39,688056 |
| 16 | 0 | 5,20 | 561 B | 18,570833 | 39,752778 |
| 17 | 0 | 4,60 | 455 B | 18,590556 | 39,722500 |
| 18 | 0 | 5,20 | 539 B | 18,590278 | 39,741111 |
| 19 | 0 | 4,50 | 526 B | 18,606667 | 39,706667 |
| 20 | 0 | 4,60 | 455 B | 18,570000 | 39,693056 |
| 21 | 0 | 4,70 | 561 B | 18,608889 | 39,735000 |
| 22 | 0 | 4,60 | 445 B | 18,575556 | 39,722500 |
| 23 | 0 | 4,60 | 455 B | 18,564444 | 39,684444 |
| 24 | 0 | 4,60 | 445 B | 18,565556 | 39,666667 |
| 25 | 0 | 4,70 | 561 B | 18,565556 | 39,704167 |
| 26 | 0 | 5,20 | 455 B | 18,566389 | 39,738333 |
| 27 | 0 | 4,70 | 445 B | 18,599722 | 39,726389 |
| 28 | 0 | 5,20 | 500 B | 18,598333 | 39,727500 |
| 29 | 0 | 5,30 | 561 M | 18,589167 | 39,690833 |
| 30 | 0 | 4,80 | 527 M | 18,573056 | 39,670556 |
| 31 | 0 | 5,30 | 533 M | 18,572500 | 39,690556 |
| 32 | 0 | 4,80 | 527 M | 18,581667 | 39,752778 |
| 33 | 0 | 4,90 | 561 M | 18,589722 | 39,717222 |
| 34 | 0 | 5,20 | 533 M | 18,522500 | 39,718056 |
| 35 | 0 | 1,90 | 561 M | 18,532222 | 39,705000 |
| 36 | 0 | 4,90 | 561 M | 18,539444 | 39,691389 |
| 37 | 0 | 4,80 | 561 M | 18,582778 | 39,703889 |
| 38 | 0 | 1,90 | 539 M | 18,599444 | 39,688056 |
| 39 | 0 | 2,30 | 539 M | 18,548333 | 39,700833 |
| 40 | 0 | 4,80 | 455 M | 18,598333 | 39,689167 |
| 41 | 0 | 1,90 | 561 M | 18,615000 | 39,709722 |
| 42 | 0 | 1,30 | 539 M | 18,531667 | 39,692222 |
| 43 | 0 | 1,90 | 561 M | 18,565000 | 39,705556 |
| 44 | 0 | 1,30 | 561 M | 18,581667 | 39,671111 |
| 45 | 0 | 1,90 | 539 M | 18,566111 | 39,719444 |
| 46 | 0 | 4,80 | 455 M | 18,599444 | 39,720000 |
| 47 | 0 | 4,80 | 455 B | 18,590278 | 39,696667 |
| 48 | 0 | 4,80 | 561 B | 18,590278 | 39,731111 |
| 49 | 0 | 5,30 | 561 B | 18,590278 | 39,718333 |

Figura 42 – Exemplo de conjunto de 50 plantios



Já para o caso dos 484 elementos só conseguimos resultado para o MST-Ci. A execução do HCS não terminou no tempo desejado.

Para este grupo ainda alteramos o valor do *threshold* para 2.17 para avaliar o comportamento. A expectativa era de criar um grafo mais esparso para o HCS executar mais rapidamente. Entretanto, depois de 8 horas não houve resultado. Com  $t=2$  ultrapassou de 24 horas sem resultado.

O MST-Ci gerou 17 clusters para o primeiro *threshold* e 32 para o segundo. O tempo de execução foi bem próximo 1:06 e 1:11, respectivamente.

Para os caso de 50 elementos, o tempo de execução foi de 3 minutos para o HCS e de poucos segundos para o MST-Ci. Já para os de 100, foi de 2 horas e de 15 minutos para o HCS e 30 seg e 8 seg para o MST-Ci.

Nos casos menores o tempo de execução foi muito próximo.

Considerando que o HCS é limitado por  $2Nxf(n,m)$  e que nosso algoritmo de *minimum cut* tem  $O(n^3)$  em tempo de execução, ficamos com um total de  $2(n^4)$ . Uma projeção em cima do tempo gasto com 50 e 100 vértices para o tempo a ser gasto com 484 não é muito animadora.

Para termos um volume maior de testes comparativos, foi criado uma bateria de entradas com quatorze conjuntos de 50 plantios e 6 conjuntos de 100 plantios (veja alguns exemplos no Capítulo 7 . Deixamos o *threshold* em 2.17 para obter um fator de similaridade melhor e só retiramos do processo vértices com grau menos que 1, ou seja, singletons já desconectados do conjunto de inicial.

Pelos gráficos apresentados a seguir, podemos perceber que o tempo de execução (Figura 43 e Figura 44) do HCS é muito maior que o MST-Ci. Enquanto este roda em poucos segundos ou menos, o HCS leva vários segundos ou até horas, para alguns casos de conjuntos com 100 elementos.

Quanto à qualidade da solução, podemos perceber nos gráficos da Figura 45 e da Figura 46, que o HCS cria um número maior de *clusters*, oferecendo ao analista um número maior de opções de planejamento. O número de singletons gerados é o mesmo pelo fato de executarmos uma adoção

final de singletons ao término do HCS, deixando isolados somente os plantios com baixa similaridade, o que também é facilmente indicado pelo MST-Ci. Um analista humano também poderia identificar alguns plantios bem diferentes do resto do conjunto e separá-los e até mesmo criar alguns grandes grupos como o MST-Ci, mas uma análise mais detalhada é bem mais complexa e o HCS conseguiu efetuar-la muito bem.

O outro fator de comparação é a similaridade interna dos *clusters* gerados. Os dois algoritmos deram resultados muito bons. O MST-Ci, gerou *clusters* com baixa similaridade em apenas cinco casos (2.156749693, 2.169645164, 2.168208, 2.148047, 2.169170), com apenas dois ruins, pois um arredondamento nos outros leva ao limite de 2.17. O HCS só teve resultados acima do *threshold*. Veja abaixo na Tabela 8 e Tabela 9.

Já a dissimilaridade (ou análise de similaridade abaixo do limite) entre os *clusters* teve um resultado melhor com o MST-Ci, apesar do HCS, em nenhum momento ter ficado acima do *threshold*. Um gráfico comparativo das similaridades é apresentado na Figura 47 e Figura 48.

**Tabela 8 - Dados dos testes nos conjuntos de 100 plantios**

| Conjunto de plantios                 | 1        | 2        | 3        | 4        | 5        | 6        |
|--------------------------------------|----------|----------|----------|----------|----------|----------|
| Tempo de execução - HCS              | 6008139  | 3777652  | 1744418  | 208249   | 910219   | 163224   |
| Tempo de execução - MST-Ci           | 2994     | 2974     | 2594     | 3505     | 2764     | 3565     |
| Nº Clusters HCS                      | 8        | 8        | 12       | 9        | 14       | 11       |
| Nº Clusters MST-Ci                   | 2        | 1        | 3        | 4        | 10       | 6        |
| Nº Singletons HCS                    | 1        | 0        | 3        | 5        | 4        | 1        |
| Nº Singletons MST-Ci                 | 1        | 0        | 3        | 5        | 4        | 1        |
| Similaridade intra-clusters - HCS    | 2,187634 | 2,186396 | 2,187106 | 2,187689 | 2,188618 | 2,186094 |
| Similaridade intra-clusters - MST-Ci | 2,177664 | 2,15675  | 2,169645 | 2,180161 | 2,178403 | 2,182335 |
| Similaridade inter-clusters - HCS    | 2,094992 | 2,047234 | 2,000973 | 1,724439 | 1,968785 | 1,866969 |
| Similaridade inter-clusters - MST-Ci | 1,986967 | NaN      | 1,967876 | 1,65199  | 1,886013 | 1,84907  |



**Tabela 9 - Dados dos testes nos conjuntos de 50 plantios**

| <b>Conjunto de plantios</b>                 | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        | 9        | 10       | 11       | 12       | 13       | 14       |
|---|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>Tempo de execução - HCS</b>              | 84141    | 46667    | 23974    | 93805    | 124218   | 59856    | 157716   | 24936    | 15572    | 4908     | 113463   | 10435    | 13900    | 1973     |
| <b>Tempo de execução - MST-Ci</b>           | 160      | 170      | 220      | 170      | 181      | 221      | 220      | 220      | 180      | 190      | 201      | 220      | 190      | 501      |
| <b>N° Clusters HCS</b>                      | 6        | 8        | 6        | 7        | 5        | 5        | 5        | 6        | 5        | 12       | 7        | 6        | 7        | 11       |
| <b>N° Clusters MST-Ci</b>                   | 2        | 4        | 1        | 3        | 2        | 2        | 1        | 2        | 3        | 9        | 3        | 4        | 3        | 7        |
| <b>N° Singletons HCS</b>                    | 2        | 3        | 9        | 1        | 3        | 0        | 1        | 2        | 3        | 4        | 0        | 1        | 0        | 13       |
| <b>N° Singletons MST-Ci</b>                 | 2        | 3        | 9        | 1        | 3        | 0        | 1        | 2        | 3        | 4        | 0        | 1        | 0        | 13       |
| <b>Similaridade intra-clusters - HCS</b>    | 2,185779 | 2,186139 | 2,182948 | 2,187466 | 2,186865 | 2,185888 | 2,186262 | 2,189317 | 2,186662 | 2,189136 | 2,189074 | 2,186069 | 2,186785 | 2,178842 |
| <b>Similaridade intra-clusters - MST-Ci</b> | 2,168208 | 2,170508 | 2,148047 | 2,179496 | 2,180277 | 2,176881 | 2,169170 | 2,177445 | 2,182902 | 2,184363 | 2,177100 | 2,182799 | 2,181790 | 2,174336 |
| <b>Similaridade inter-clusters - HCS</b>    | 2,070967 | 2,047430 | 2,039420 | 2,077160 | 2,101975 | 2,070398 | 2,106399 | 2,047072 | 1,588226 | 1,929041 | 2,068711 | 1,969786 | 1,874523 | 1,830603 |
| <b>Similaridade inter-clusters - MST-Ci</b> | 1,978831 | 1,975057 | 1,971177 | 1,983188 | 1,977113 | 1,991924 | 1,985890 | 1,992893 | 1,503819 | 1,908159 | 1,811931 | 1,960895 | 1,817849 | 1,821916 |



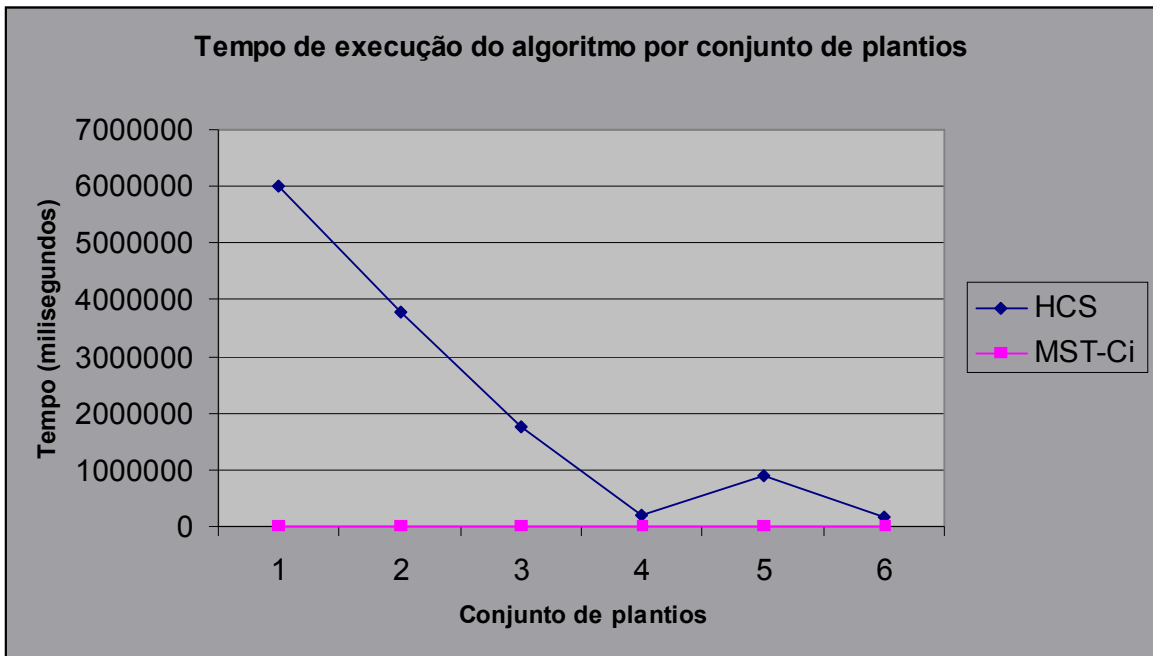


Figura 43 – Gráfico de tempo de execução para conjuntos de 100 plantios

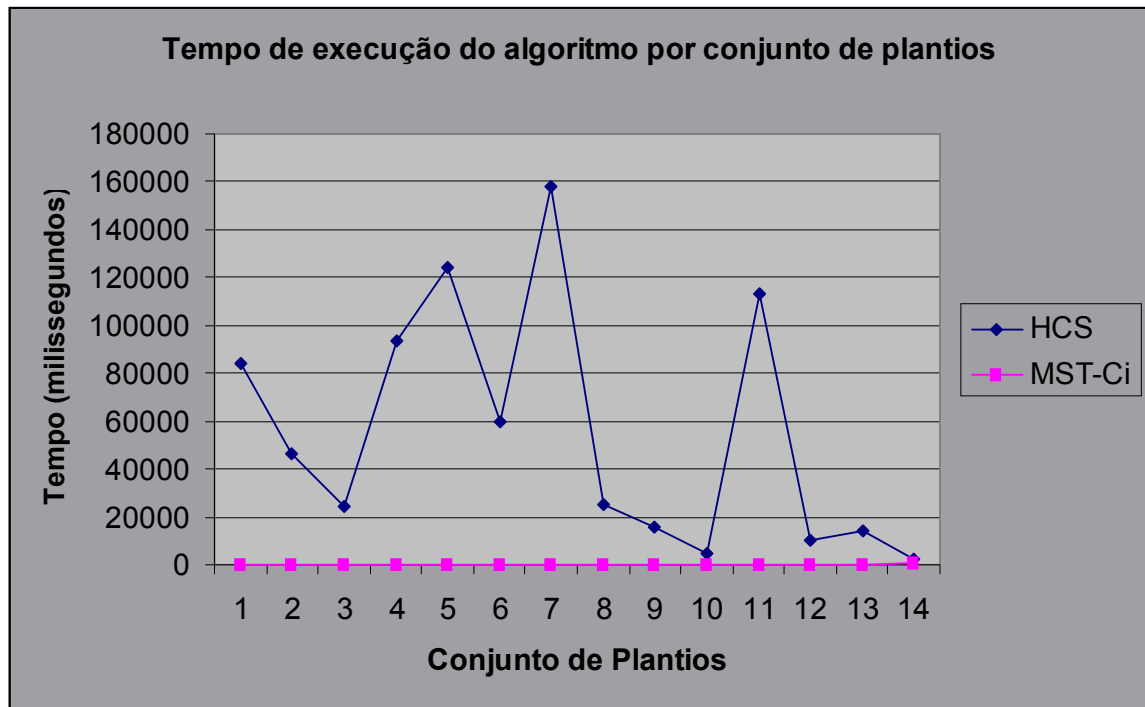


Figura 44 - Gráfico de tempo de execução para conjuntos de 50 plantios



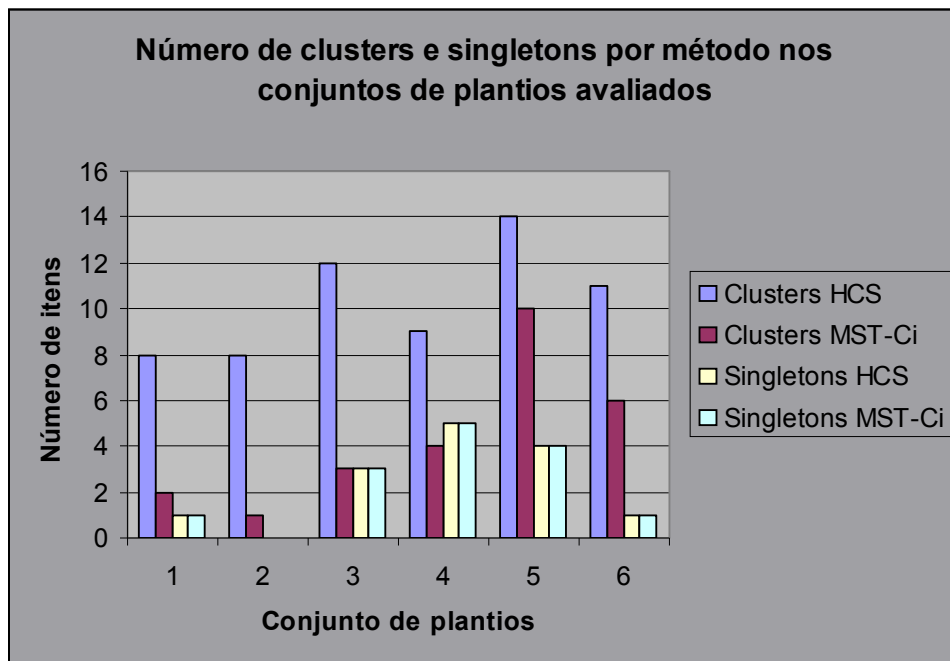


Figura 45 - Gráfico do número de *clusters* e *singletons* gerados para conjuntos de 100 plantios

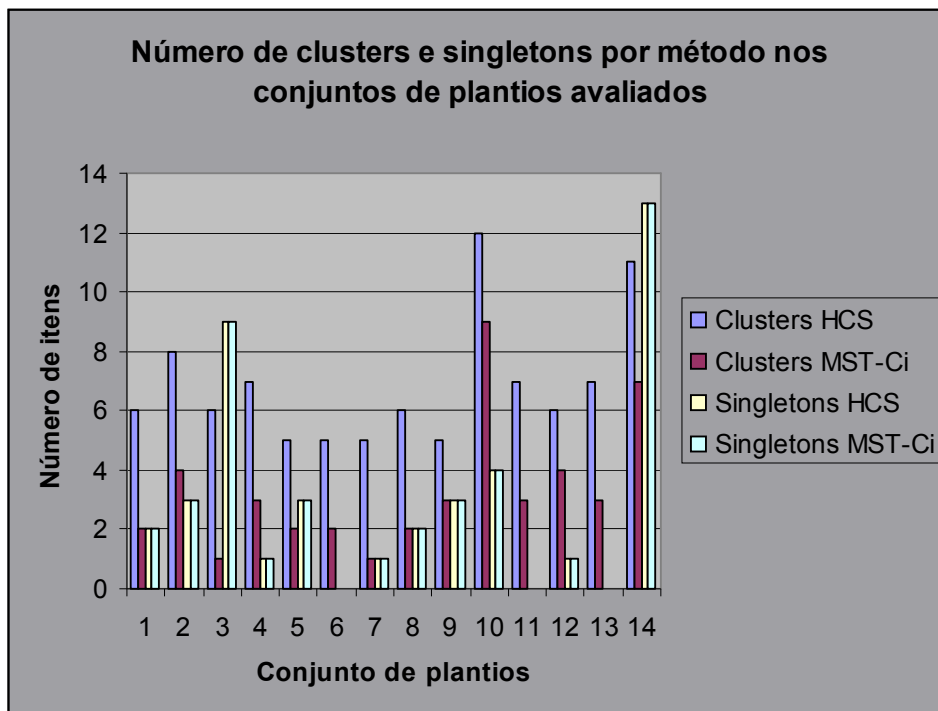


Figura 46 - Gráfico do número de *clusters* e *singletons* gerados para conjuntos de 50 plantios

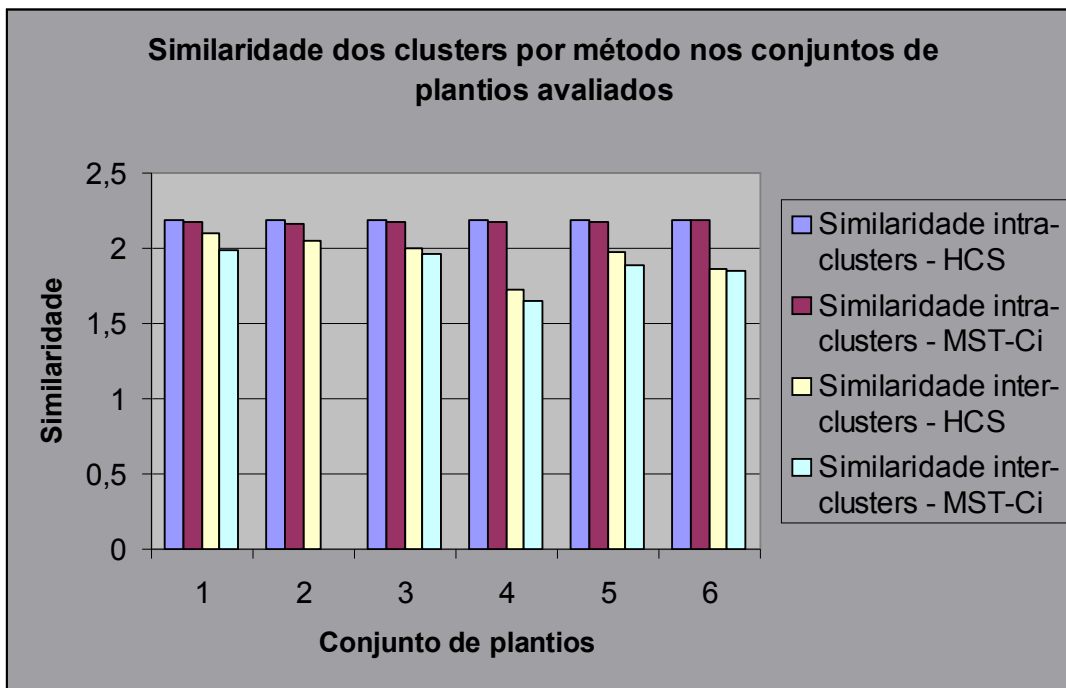


Figura 47 - Gráfico de similaridade dos *clusters* gerados para conjuntos de 100 plantios

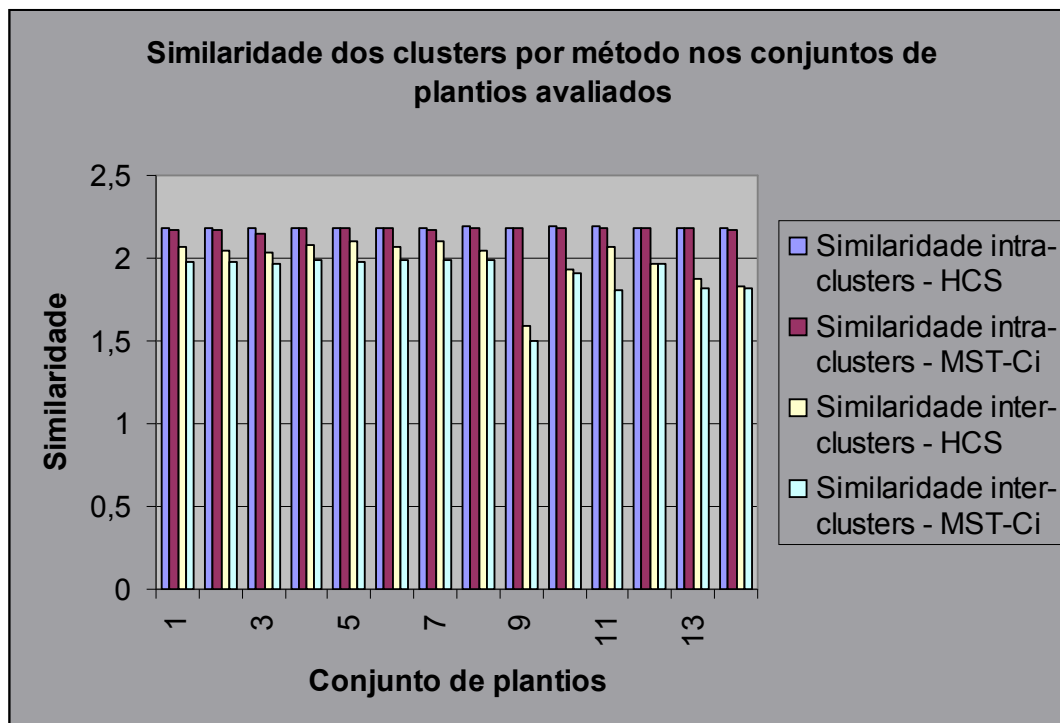


Figura 48 - Gráfico de similaridade dos *clusters* gerados para conjuntos de 50 plantios

## Capítulo 6 Conclusões

Poderíamos ter alcançado mais resultados de estudo com o uso de uma máquina mais potente, mas o que foi realizado já conduziu a conclusões interessantes. Um algoritmo de *minimum cut* menos complexo também traria um ganho na execução. Hartuv e Shamir afirmam que existem algoritmos da ordem de  $O(nm)$  contra nosso  $O(n^3)$ .

Com o estudo apresentado pudemos observar que a complexidade e o volume de informações a serem avaliadas nos colocam no dilema de buscar uma solução rápida e eficiente. Algoritmos hierárquicos deixam a desejar na qualidade da solução e os de particionamento são vinculados a um número pré-definido de clusters e como o algoritmo em teoria dos grafos HCS, têm um processamento mais pesado.

A forma como se usa as características dos elementos a serem agrupados também é muito importante. Se criarmos classes de características criaremos uma separação forte antes da execução dos algoritmos. Por exemplo, se usássemos a classe de densidade de forma exclusiva, 1 não se junta a 2,3 ou 4 e da mesma forma as outras se excluíssem, já teríamos subgrafos originados apenas da análise dessa característica. Depois, o algoritmo criaria clusters dentro desses subgrafos. Se o volume for muito grande, pode ser uma técnica interessante, mas deve-se considerar a perda de continuidade, já que elementos com densidades próximas, mas no limite de duas classes, podem ficar separados.

Uma outra variação é ser mais ou menos exigente nos critérios. Podemos definir um *threshold* maior para a similaridade e gerar clusters menores ou o contrário. Lembramos que o *threshold* significa um limite(marco) para termos uma similaridade boa entre elementos.

A implementação do MST-Ci se mostrou muito prática e rápida, embora a qualidade da solução seja questionável em alguns pontos de comparação com a solução do HCS, como o fato de manter num mesmo grupo elementos com similaridade menor que o *threshold* definido e não ser capaz de ir a fundo na criação de grupos menores e mais coesos. Como comportamento humano, é bem semelhante, inclusive pela proposta de suporte com MST de reduzir a complexidade da computação

e simplificar a implementação, ou seja, do ser humano fazer. Fica a certeza de que o HCS traz soluções muito boas e devemos buscar algoritmos de apoio mais eficientes.

Podemos perceber nos resultados, que o MST-Ci normalmente cria grupos maiores que o HCS. Os singletons são agrupados no HCS, o que nos dá um resultado parecido. Quanto aos grupos maiores, existem as ligações fracas que o HCS não mantém. Assim, uma alternativa para melhorar o resultado seria tentar quebrar num nível maior o grafo para executar o HCS para esses subgrupos. Isso reduziria a complexidade da execução total.

## Referências bibliográficas

- Ahuja, R. K.; Magnanti, T. L. and Orlin, J. B. (1993), Network flows: theory, algorithms, and applications, Prentice Hall, Inc. 166-293
- Ben-Dor, A., Shamir R. and Yakhini Z. (1999), Clustering gene expression patterns, Journal of Computational Biology, 6, 281-297
- Cook, W. J.; Cunningham, W. H.; Pulleyblank, W. R. and Schrijver, A. (1998), Combinatorial optimization, Wiley-Interscience series in discrete mathematics and optimization 37-76
- Hansen, P. and Jaumard, B. (1997), Cluster analysis and mathematical programming, Mathematical Programming, 79, 191-215
- Hartuv, E. and Shamir, R. (2000), A clustering algorithm based on graph connectivity, Information Processing Letters 76 175-181
- Herwig, R.; Poustka, A. J.; Mueller, C.; Lehrach, H. and O'Brien, J. (1999), Large-scale clustering of cDNA-fingerprinting data. *Genome Research*, 9(11):1093-1105, November 1999
- Kljajnsek, G., Merging a set of polygons with non-stable borders, laboratory for Geometric Modeling and Multimedia Algorithms –University of Maribor/Slovenia
- Kohonen, T. (1990), The self organizing map, Proceedings of the IEEE, 78, 1464-1480
- MacQueen, J. (1965), Some methods for classification and analysis of multivariate observations, Proceedings of the 5<sup>th</sup> Berkeley Symposium on Mathematical Statistics and Probability, 281-297
- Matula, D. W., K (1972)-Components, clusters and slicings in graphs, SIAM J. Appl. Math. 22(3) 459-480
- Michailidis, G. (2000) – Stat 700 – Cluster Analysis (Unsupervised Learning), Lecture Notes, University of Michigan
- Shamir, S. and Sharan, R. (2000), Algorithmic approaches to clustering gene expression data, Technical Report, Dept of CS, Tel-Aviv University

Sharan, R. and Shamir, R.(2000), CLICK: A clustering algorithm with applications to gene expression analysis, Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB), 307-316

Tavakoly, B. (2003), Gene Expression Data Clustering With Minimum Spanning Tree (MST) - Department of Information Systems and Computing, Brunel University

Wu, Z. and Leahy, R. (1993), An optimal graph theoretic approach to data clustering: theory and its application to image segmentation, IEEE Trans. Pattern Analysis Machine Intelligence 15(11) 1101-1113

## **Capítulo 7 Apêndice A – Exemplos de conjuntos de testes**

Seguem 3 conjuntos de plantios que foram usados nos testes e o resultado da aplicação dos algoritmos sobre seus dados.

Uma observação importante é que ao final de cada tabela existem alguns itens com identificação (coluna ID) 9999. Estes itens servem para indicar o centróide, ou plantio de característica média, do grupo ao qual ele se refere na sua coluna GRUPO.

Tabela 10 – Resultado do HCS sobre o conjunto 1 de 50 plantios

| ID   | GRUPO | DENSIDADE | PRODUTIV | IDADE       | LATITUDE  | LONGITUDE | SIMILARIDADE |
|------|-------|-----------|----------|-------------|-----------|-----------|--------------|
| 52   | 0     | 539       | B        | 1,8         | 18,573611 | 39,675556 | 2,174825     |
| 53   | 0     | 539       | B        | 1,8         | 18,556944 | 39,686944 | 2,185856     |
| 54   | 0     | 539       | B        | 1,8         | 18,553889 | 39,704722 | 2,193220     |
| 59   | 0     | 539       | B        | 1,6         | 18,587222 | 39,715556 | 2,176161     |
| 55   | 0     | 539       | B        | 1,8         | 18,537222 | 39,732500 | 2,167760     |
| 66   | 1     | 561       | B        | 4,6         | 18,591111 | 39,730833 | 2,185460     |
| 67   | 1     | 509       | B        | 4,6         | 18,591389 | 39,736667 | 2,181200     |
| 69   | 1     | 561       | B        | 4,6         | 18,574722 | 39,688889 | 2,175807     |
| 70   | 1     | 509       | B        | 4,3         | 18,608056 | 39,737500 | 2,171267     |
| 71   | 1     | 509       | B        | 4,6         | 18,574722 | 39,729722 | 2,188207     |
| 72   | 1     | 445       | B        | 4,3         | 18,558056 | 39,698889 | 2,175604     |
| 74   | 1     | 445       | B        | 4,3         | 18,558056 | 39,706944 | 2,179158     |
| 75   | 1     | 561       | B        | 4,3         | 18,558056 | 39,737222 | 2,174493     |
| 76   | 1     | 509       | B        | 4,3         | 18,583889 | 39,720278 | 2,196171     |
| 77   | 1     | 561       | B        | 4,6         | 18,585833 | 39,720833 | 2,194625     |
| 78   | 1     | 455       | B        | 4,6         | 18,591389 | 39,688056 | 2,174027     |
| 82   | 1     | 509       | B        | 4,3         | 18,583889 | 39,723333 | 2,194060     |
| 86   | 1     | 445       | B        | 4,4         | 18,567222 | 39,701111 | 2,182462     |
| 89   | 1     | 509       | B        | 4,4         | 18,589722 | 39,688333 | 2,174701     |
| 90   | 1     | 561       | B        | 4,3         | 18,606944 | 39,709722 | 2,177086     |
| 92   | 1     | 445       | B        | 4,4         | 18,556944 | 39,707500 | 2,178480     |
| 94   | 1     | 561       | B        | 4,4         | 18,557222 | 39,723889 | 2,179438     |
| 95   | 1     | 509       | B        | 4,4         | 18,590278 | 39,719444 | 2,191690     |
| 97   | 1     | 561       | B        | 4,3         | 18,586944 | 39,718056 | 2,194675     |
| 98   | 1     | 561       | B        | 4,3         | 18,586944 | 39,721667 | 2,193457     |
| 99   | 1     | 575       | B        | 4,3         | 18,591944 | 39,708056 | 2,187897     |
| 81   | 1     | 561       | B        | 4,3         | 18,581944 | 39,757778 | 2,165722     |
| 58   | 2     | 561       | B        | 4,8         | 18,570556 | 39,733889 | 2,184278     |
| 57   | 2     | 455       | B        | 4,8         | 18,553889 | 39,756667 | 2,189531     |
| 65   | 2     | 561       | B        | 4,6         | 18,571667 | 39,765000 | 2,187741     |
| 73   | 3     | 561       | B        | 4,6         | 18,566111 | 39,680833 | 2,191675     |
| 79   | 3     | 575       | B        | 4,6         | 18,574722 | 39,669444 | 2,191453     |
| 80   | 3     | 445       | B        | 4,3         | 18,567222 | 39,689167 | 2,189085     |
| 87   | 3     | 561       | B        | 4,4         | 18,589722 | 39,683333 | 2,187968     |
| 93   | 3     | 561       | B        | 4,2         | 18,570833 | 39,670556 | 2,191318     |
| 96   | 3     | 445       | B        | 4,3         | 18,586944 | 39,683889 | 2,190038     |
| 50   | 4     | 561       | B        | 4,8         | 18,590278 | 39,705000 | 2,194391     |
| 61   | 4     | 561       | B        | 4,8         | 18,604444 | 39,707222 | 2,191539     |
| 68   | 4     | 556       | B        | 5           | 18,608056 | 39,702778 | 2,189866     |
| 60   | 4     | 561       | B        | 4,8         | 18,603889 | 39,698333 | 2,192841     |
| 51   | 4     | 539       | B        | 5,3         | 18,573611 | 39,695000 | 2,180312     |
| 63   | 5     | 445       | B        | 4,8         | 18,537778 | 39,682222 | 2,184447     |
| 64   | 5     | 509       | B        | 4,6         | 18,521944 | 39,697778 | 2,192407     |
| 83   | 5     | 445       | B        | 4,3         | 18,523056 | 39,726389 | 2,176353     |
| 84   | 5     | 561       | B        | 4,4         | 18,532778 | 39,701667 | 2,197466     |
| 85   | 5     | 445       | B        | 4,4         | 18,534722 | 39,689167 | 2,190822     |
| 88   | 5     | 445       | B        | 4,4         | 18,542222 | 39,704167 | 2,189607     |
| 91   | 5     | 445       | B        | 4,3         | 18,523333 | 39,694167 | 2,192326     |
| 56   | 6     | 548       | B        | 5,3         | 18,537222 | 39,749722 | 2,200000     |
| 62   | 7     | 433       | B        | 5,3         | 18,586667 | 39,764167 | 2,200000     |
| 9999 | 0     | 539       | B        | 1,76        | 18,561778 | 39,703056 | 0,000000     |
| 9999 | 1     | 519       | B        | 4,404545455 | 18,580694 | 39,717033 | 0,000000     |
| 9999 | 2     | 525       | B        | 4,733333333 | 18,565370 | 39,751852 | 0,000000     |
| 9999 | 3     | 524       | B        | 4,4         | 18,575926 | 39,679537 | 0,000000     |
| 9999 | 4     | 555       | B        | 4,94        | 18,596056 | 39,701667 | 0,000000     |
| 9999 | 5     | 470       | B        | 4,457142857 | 18,530833 | 39,699365 | 0,000000     |
| 9999 | 6     | 548       | B        | 5,3         | 18,537222 | 39,749722 | 0,000000     |
| 9999 | 7     | 433       | B        | 5,3         | 18,586667 | 39,764167 | 0,000000     |



Tabela 11 – Resultado do MST-Ci sobre o conjunto 1 de 50 plantios

| ID   | GRUPO | DENSIDADE | PRODUTIV | IDADE | LATITUDE   | LONGITUDE  | SIMILARIDADE |
|------|-------|-----------|----------|-------|------------|------------|--------------|
| 56   | 0     | 548       | B        | 5,3   | 18,5372222 | 39,7497222 | 2,2000000    |
| 62   | 1     | 433       | B        | 5,3   | 18,5866667 | 39,7641667 | 2,2000000    |
| 52   | 2     | 539       | B        | 1,8   | 18,5736111 | 39,6755556 | 2,1748253    |
| 53   | 2     | 539       | B        | 1,8   | 18,5569444 | 39,6869444 | 2,1858557    |
| 54   | 2     | 539       | B        | 1,8   | 18,5538889 | 39,7047222 | 2,1932198    |
| 55   | 2     | 539       | B        | 1,8   | 18,5372222 | 39,7325000 | 2,1677601    |
| 59   | 2     | 539       | B        | 1,6   | 18,5872222 | 39,7155556 | 2,1761614    |
| 51   | 3     | 539       | B        | 5,3   | 18,5736111 | 39,6950000 | 1,9877227    |
| 83   | 3     | 445       | B        | 4,3   | 18,5230556 | 39,7263889 | 2,1559791    |
| 90   | 3     | 561       | B        | 4,3   | 18,6069444 | 39,7097222 | 2,1711441    |
| 50   | 3     | 561       | B        | 4,8   | 18,5902778 | 39,7050000 | 2,1846705    |
| 60   | 3     | 561       | B        | 4,8   | 18,6038889 | 39,6983333 | 2,1720680    |
| 61   | 3     | 561       | B        | 4,8   | 18,6044444 | 39,7072222 | 2,1731741    |
| 68   | 3     | 556       | B        | 5     | 18,6080556 | 39,7027778 | 1,9696680    |
| 58   | 3     | 561       | B        | 4,8   | 18,5705556 | 39,7338889 | 2,1794738    |
| 71   | 3     | 509       | B        | 4,6   | 18,5747222 | 39,7297222 | 2,1829606    |
| 66   | 3     | 561       | B        | 4,6   | 18,5911111 | 39,7308333 | 2,1763078    |
| 67   | 3     | 509       | B        | 4,6   | 18,5913889 | 39,7366667 | 2,1722847    |
| 95   | 3     | 509       | B        | 4,4   | 18,5902778 | 39,7194444 | 2,1829932    |
| 76   | 3     | 509       | B        | 4,3   | 18,5838889 | 39,7202778 | 2,1869323    |
| 77   | 3     | 561       | B        | 4,6   | 18,5858333 | 39,7208333 | 2,1854043    |
| 98   | 3     | 561       | B        | 4,3   | 18,5869444 | 39,7216667 | 2,1842385    |
| 97   | 3     | 561       | B        | 4,3   | 18,5869444 | 39,7180556 | 2,1860052    |
| 82   | 3     | 509       | B        | 4,3   | 18,5838889 | 39,7233333 | 2,1850451    |
| 99   | 3     | 575       | B        | 4,3   | 18,5919444 | 39,7080556 | 2,1837082    |
| 63   | 3     | 445       | B        | 4,8   | 18,5377778 | 39,6822222 | 2,1627494    |
| 85   | 3     | 445       | B        | 4,4   | 18,5347222 | 39,6891667 | 2,1638012    |
| 64   | 3     | 509       | B        | 4,6   | 18,5219444 | 39,6977778 | 2,1562416    |
| 91   | 3     | 445       | B        | 4,3   | 18,5233333 | 39,6941667 | 2,1565715    |
| 84   | 3     | 561       | B        | 4,4   | 18,5327778 | 39,7016667 | 2,1658369    |
| 88   | 3     | 445       | B        | 4,4   | 18,5422222 | 39,7041667 | 2,1740307    |
| 73   | 3     | 561       | B        | 4,6   | 18,5661111 | 39,6808333 | 2,1752239    |
| 69   | 3     | 561       | B        | 4,6   | 18,5747222 | 39,6888889 | 2,1825229    |
| 80   | 3     | 445       | B        | 4,3   | 18,5672222 | 39,6891667 | 2,1822530    |
| 79   | 3     | 575       | B        | 4,6   | 18,5747222 | 39,6694444 | 2,1662151    |
| 93   | 3     | 561       | B        | 4,2   | 18,5708333 | 39,6705556 | 2,1671605    |
| 72   | 3     | 445       | B        | 4,3   | 18,5580556 | 39,6988889 | 2,1848075    |
| 74   | 3     | 445       | B        | 4,3   | 18,5580556 | 39,7069444 | 2,1875482    |
| 92   | 3     | 445       | B        | 4,4   | 18,5569444 | 39,7075000 | 2,1866967    |
| 86   | 3     | 445       | B        | 4,4   | 18,5672222 | 39,7011111 | 2,1915600    |
| 87   | 3     | 561       | B        | 4,4   | 18,5897222 | 39,6833333 | 2,1736706    |
| 96   | 3     | 445       | B        | 4,3   | 18,5869444 | 39,6838889 | 2,1752785    |
| 78   | 3     | 455       | B        | 4,6   | 18,5913889 | 39,6880556 | 2,1759987    |
| 89   | 3     | 509       | B        | 4,4   | 18,5897222 | 39,6883333 | 2,1770781    |
| 75   | 3     | 561       | B        | 4,3   | 18,5580556 | 39,7372222 | 2,1737135    |
| 94   | 3     | 561       | B        | 4,4   | 18,5572222 | 39,7238889 | 2,1823114    |
| 70   | 3     | 509       | B        | 4,3   | 18,6080556 | 39,7375000 | 2,1620638    |
| 57   | 3     | 455       | B        | 4,8   | 18,5538889 | 39,7566667 | 2,1573743    |
| 65   | 3     | 561       | B        | 4,6   | 18,5716667 | 39,7650000 | 2,1533797    |
| 81   | 3     | 561       | B        | 4,3   | 18,5819444 | 39,7577778 | 2,1587101    |
| 9999 | 0     | 548       | B        | 5,3   | 18,5372222 | 39,7497222 | 0,0000000    |
| 9999 | 1     | 433       | B        | 5,3   | 18,5866667 | 39,7641667 | 0,0000000    |
| 9999 | 2     | 539       | B        | 2     | 18,5617778 | 39,7030556 | 0,0000000    |
| 9999 | 3     | 516       | B        | 4     | 18,5726292 | 39,7095672 | 0,0000000    |

Tabela 12 - Resultado do HCS sobre o conjunto 2 de 50 plantios

| ID   | GRUPO | DENSIDADE | PRODUTIV | IDADE | LATITUDE    | LONGITUDE   | SIMILARIDADE |             |
|------|-------|-----------|----------|-------|-------------|-------------|--------------|-------------|
| 104  | 0     | 561       | B        |       | 5           | 18,55694444 | 39,72638889  | 2,185638485 |
| 106  | 0     | 561       | B        |       | 5           | 18,53916667 | 39,75555556  | 2,185638485 |
| 108  | 1     | 500       | B        |       | 12,9        | 18,57388889 | 39,70277778  | 2,185845122 |
| 140  | 1     | 500       | B        |       | 12,8        | 18,60388889 | 39,6875      | 2,185845122 |
| 135  | 2     | 500       | B        |       | 12,8        | 18,54027778 | 39,71555556  | 2,190024745 |
| 133  | 2     | 500       | B        |       | 12,8        | 18,52305556 | 39,70416667  | 2,186213578 |
| 138  | 2     | 500       | B        |       | 12,8        | 18,53722222 | 39,68722222  | 2,185999937 |
| 142  | 2     | 500       | B        |       | 12,8        | 18,55666667 | 39,66944444  | 2,167742318 |
| 105  | 2     | 561       | B        |       | 12,8        | 18,54       | 39,74222222  | 2,167622119 |
| 100  | 3     | 445       | B        |       | 4,3         | 18,58694444 | 39,68722222  | 2,18214259  |
| 102  | 3     | 509       | B        |       | 4,3         | 18,57527778 | 39,68416667  | 2,17771256  |
| 103  | 3     | 445       | B        |       | 4,3         | 18,55694444 | 39,70583333  | 2,175524573 |
| 115  | 3     | 509       | B        |       | 4,5         | 18,56805556 | 39,72555556  | 2,179184001 |
| 116  | 3     | 509       | B        |       | 4,3         | 18,59944444 | 39,73694444  | 2,173470439 |
| 117  | 3     | 561       | B        |       | 4,5         | 18,59944444 | 39,70472222  | 2,188218367 |
| 118  | 3     | 445       | B        |       | 4,3         | 18,60416667 | 39,69916667  | 2,18280151  |
| 120  | 3     | 509       | B        |       | 4,5         | 18,60416667 | 39,72333333  | 2,180200548 |
| 126  | 3     | 561       | B        |       | 4,4         | 18,59055556 | 39,73222222  | 2,179622896 |
| 127  | 3     | 561       | B        |       | 4,5         | 18,59055556 | 39,69777778  | 2,190232636 |
| 129  | 3     | 509       | B        |       | 4,5         | 18,57388889 | 39,6975      | 2,186320169 |
| 131  | 3     | 526       | B        |       | 4,5         | 18,58277778 | 39,71944444  | 2,190368665 |
| 132  | 3     | 445       | B        |       | 4,3         | 18,58972222 | 39,72222222  | 2,187976252 |
| 136  | 3     | 445       | B        |       | 4,4         | 18,57361111 | 39,6975      | 2,186146326 |
| 137  | 3     | 509       | B        |       | 4,3         | 18,59027778 | 39,70694444  | 2,196138715 |
| 139  | 3     | 455       | B        |       | 4,4         | 18,58722222 | 39,70361111  | 2,195802066 |
| 143  | 3     | 575       | B        |       | 4,4         | 18,57361111 | 39,7225      | 2,184274904 |
| 146  | 3     | 445       | B        |       | 4,4         | 18,59333333 | 39,72638889  | 2,183670346 |
| 147  | 3     | 561       | B        |       | 4,5         | 18,59583333 | 39,71833333  | 2,188228054 |
| 148  | 3     | 509       | B        |       | 4,6         | 18,59055556 | 39,70583333  | 2,195540798 |
| 149  | 3     | 509       | B        |       | 4,6         | 18,59333333 | 39,68444444  | 2,178889306 |
| 121  | 3     | 561       | B        |       | 4,5         | 18,57083333 | 39,68388889  | 2,175766424 |
| 109  | 4     | 575       | B        |       | 5           | 18,59027778 | 39,68666667  | 2,194984777 |
| 112  | 4     | 561       | B        |       | 4,9         | 18,59055556 | 39,66833333  | 2,181756635 |
| 110  | 4     | 575       | B        |       | 5           | 18,60583333 | 39,71333333  | 2,178131656 |
| 113  | 5     | 561       | B        |       | 4,9         | 18,54583333 | 39,68833333  | 2,188056552 |
| 122  | 5     | 509       | B        |       | 4,5         | 18,55722222 | 39,68222222  | 2,198596345 |
| 101  | 5     | 445       | B        |       | 4,3         | 18,57027778 | 39,67138889  | 2,186919156 |
| 107  | 6     | 505       | B        |       | 4,4         | 18,55638889 | 39,73777778  | 2,198428738 |
| 119  | 6     | 509       | B        |       | 4,5         | 18,57083333 | 39,74972222  | 2,185791264 |
| 123  | 6     | 509       | B        |       | 4,5         | 18,55722222 | 39,715       | 2,179675317 |
| 124  | 6     | 445       | B        |       | 4,5         | 18,55722222 | 39,74055556  | 2,19878066  |
| 125  | 6     | 455       | B        |       | 4,4         | 18,55722222 | 39,73083333  | 2,192983773 |
| 130  | 6     | 455       | B        |       | 4,8         | 18,58277778 | 39,75333333  | 2,175735246 |
| 144  | 6     | 509       | B        |       | 4,3         | 18,55055556 | 39,72444444  | 2,186261763 |
| 114  | 6     | 561       | B        |       | 4,5         | 18,52888889 | 39,76166667  | 2,169300801 |
| 134  | 7     | 445       | B        |       | 4,4         | 18,52555556 | 39,69638889  | 2,196373827 |
| 141  | 7     | 561       | B        |       | 4,4         | 18,52333333 | 39,70472222  | 2,196373827 |
| 111  | 8     | 575       | B        |       | 12,8        | 18,60694444 | 39,76444444  | 2,2         |
| 128  | 9     | 455       | B        |       | 13,4        | 18,59055556 | 39,68166667  | 2,2         |
| 145  | 10    | 569       | B        |       | 0           | 18,59333333 | 39,69        | 2,2         |
| 9999 | 0     | 561       | B        |       | 5           | 18,54805555 | 39,74097223  | 0           |
| 9999 | 1     | 500       | B        |       | 12,85       | 18,58888889 | 39,69513889  | 0           |
| 9999 | 2     | 512       | B        |       | 12,8        | 18,53944445 | 39,70372222  | 0           |
| 9999 | 3     | 504       | B        |       | 4,422727273 | 18,58593434 | 39,70843434  | 0           |
| 9999 | 4     | 570       | B        |       | 4,966666667 | 18,59555556 | 39,68944444  | 0           |
| 9999 | 5     | 505       | B        |       | 4,566666667 | 18,55777778 | 39,68064815  | 0           |
| 9999 | 6     | 493       | B        |       | 4,4875      | 18,55763889 | 39,73916667  | 0           |
| 9999 | 7     | 503       | B        |       | 4,4         | 18,52444445 | 39,70055556  | 0           |
| 9999 | 8     | 575       | B        |       | 12,8        | 18,60694444 | 39,76444444  | 0           |
| 9999 | 9     | 455       | B        |       | 13,4        | 18,59055556 | 39,68166667  | 0           |
| 9999 | 10    | 569       | B        |       | 0           | 18,59333333 | 39,69        | 0           |

Tabela 13 - Resultado do MST-Ci sobre o conjunto 2 de 50 plantios

| ID   | GRUPO | DENSIDADE | PRODUTIV | IDADE | LATITUDE  | LONGITUDE | SIMILARIDADE |
|------|-------|-----------|----------|-------|-----------|-----------|--------------|
| 111  | 0     | 575       | B        | 12,8  | 18,606944 | 39,764444 | 2,200000     |
| 128  | 1     | 455       | B        | 13,4  | 18,590556 | 39,681667 | 2,200000     |
| 145  | 2     | 569       | B        | 0     | 18,593333 | 39,690000 | 2,200000     |
| 105  | 3     | 561       | B        | 12,8  | 18,540000 | 39,742222 | 2,167622     |
| 133  | 3     | 500       | B        | 12,8  | 18,523056 | 39,704167 | 2,186214     |
| 135  | 3     | 500       | B        | 12,8  | 18,540278 | 39,715556 | 2,190025     |
| 138  | 3     | 500       | B        | 12,8  | 18,537222 | 39,687222 | 2,186000     |
| 142  | 3     | 500       | B        | 12,8  | 18,556667 | 39,669444 | 2,167742     |
| 108  | 4     | 500       | B        | 12,9  | 18,573889 | 39,702778 | 2,185845     |
| 140  | 4     | 500       | B        | 12,8  | 18,603889 | 39,687500 | 2,185845     |
| 104  | 5     | 561       | B        | 5     | 18,556944 | 39,726389 | 2,185638     |
| 106  | 5     | 561       | B        | 5     | 18,539167 | 39,755556 | 2,185638     |
| 114  | 6     | 561       | B        | 4,5   | 18,528889 | 39,761667 | 2,142108     |
| 110  | 6     | 575       | B        | 5     | 18,605833 | 39,713333 | 1,974218     |
| 119  | 6     | 509       | B        | 4,5   | 18,570833 | 39,749722 | 2,167055     |
| 130  | 6     | 455       | B        | 4,8   | 18,582778 | 39,753333 | 2,163679     |
| 112  | 6     | 561       | B        | 4,9   | 18,590556 | 39,668333 | 2,162052     |
| 101  | 6     | 445       | B        | 4,3   | 18,570278 | 39,671389 | 2,166596     |
| 100  | 6     | 445       | B        | 4,3   | 18,586944 | 39,687222 | 2,177882     |
| 109  | 6     | 575       | B        | 5     | 18,590278 | 39,686667 | 1,976108     |
| 149  | 6     | 509       | B        | 4,6   | 18,593333 | 39,684444 | 2,173138     |
| 117  | 6     | 561       | B        | 4,5   | 18,599444 | 39,704722 | 2,179046     |
| 118  | 6     | 445       | B        | 4,3   | 18,604167 | 39,699167 | 2,173813     |
| 127  | 6     | 561       | B        | 4,5   | 18,590556 | 39,697778 | 2,183121     |
| 139  | 6     | 455       | B        | 4,4   | 18,587222 | 39,703611 | 2,188279     |
| 137  | 6     | 509       | B        | 4,3   | 18,590278 | 39,706944 | 2,186977     |
| 148  | 6     | 509       | B        | 4,6   | 18,590556 | 39,705833 | 2,186492     |
| 102  | 6     | 509       | B        | 4,3   | 18,575278 | 39,684167 | 2,177607     |
| 121  | 6     | 561       | B        | 4,5   | 18,570833 | 39,683889 | 2,177067     |
| 116  | 6     | 509       | B        | 4,3   | 18,599444 | 39,736944 | 2,170060     |
| 120  | 6     | 509       | B        | 4,5   | 18,604167 | 39,723333 | 2,173519     |
| 131  | 6     | 526       | B        | 4,5   | 18,582778 | 39,719444 | 2,190374     |
| 126  | 6     | 561       | B        | 4,4   | 18,590556 | 39,732222 | 2,177872     |
| 132  | 6     | 445       | B        | 4,3   | 18,589722 | 39,722222 | 2,184513     |
| 146  | 6     | 445       | B        | 4,4   | 18,593333 | 39,726389 | 2,179939     |
| 147  | 6     | 561       | B        | 4,5   | 18,595833 | 39,718333 | 2,181590     |
| 115  | 6     | 509       | B        | 4,5   | 18,568056 | 39,725556 | 2,186183     |
| 143  | 6     | 575       | B        | 4,4   | 18,573611 | 39,722500 | 2,190060     |
| 103  | 6     | 445       | B        | 4,3   | 18,556944 | 39,705833 | 2,184029     |
| 123  | 6     | 509       | B        | 4,5   | 18,557222 | 39,715000 | 2,184411     |
| 107  | 6     | 505       | B        | 4,4   | 18,556389 | 39,737778 | 2,172304     |
| 124  | 6     | 445       | B        | 4,5   | 18,557222 | 39,740556 | 2,170730     |
| 125  | 6     | 455       | B        | 4,4   | 18,557222 | 39,730833 | 2,177320     |
| 144  | 6     | 509       | B        | 4,3   | 18,550556 | 39,724444 | 2,176254     |
| 129  | 6     | 509       | B        | 4,5   | 18,573889 | 39,697500 | 2,188758     |
| 136  | 6     | 445       | B        | 4,4   | 18,573611 | 39,697500 | 2,188731     |
| 113  | 6     | 561       | B        | 4,9   | 18,545833 | 39,688333 | 2,168858     |
| 122  | 6     | 509       | B        | 4,5   | 18,557222 | 39,682222 | 2,171577     |
| 134  | 6     | 445       | B        | 4,4   | 18,525556 | 39,696389 | 2,156469     |
| 141  | 6     | 561       | B        | 4,4   | 18,523333 | 39,704722 | 2,156022     |
| 9999 | 0     | 575       | B        | 12,8  | 18,606944 | 39,764444 | 0,000000     |
| 9999 | 1     | 455       | B        | 13,4  | 18,590556 | 39,681667 | 0,000000     |
| 9999 | 2     | 569       | B        | 0     | 18,593333 | 39,690000 | 0,000000     |
| 9999 | 3     | 512       | B        | 12,8  | 18,539444 | 39,703722 | 0,000000     |
| 9999 | 4     | 500       | B        | 13    | 18,588889 | 39,695139 | 0,000000     |
| 9999 | 5     | 561       | B        | 5     | 18,548056 | 39,740972 | 0,000000     |
| 9999 | 6     | 507       | B        | 4     | 18,575278 | 39,710797 | 0,000000     |

Tabela 14 - Resultado do HCS sobre o conjunto 3 de 50 plantios

| ID   | GRUPO | DENSIDADE | PRODUTIV | IDADE       | LATITUDE    | LONGITUDE   | SIMILARIDADE |
|------|-------|-----------|----------|-------------|-------------|-------------|--------------|
| 198  | 0     | 556       | M        | 4           | 18,59027778 | 39,69888889 | 1,992587556  |
| 195  | 0     | 539       | B        | 3,8         | 18,59027778 | 39,72222222 | 2,187024021  |
| 196  | 0     | 561       | B        | 3,8         | 18,59027778 | 39,72083333 | 2,188164771  |
| 197  | 0     | 556       | M        | 4,3         | 18,59027778 | 39,71305556 | 2,194344209  |
| 199  | 0     | 539       | M        | 4,3         | 18,59027778 | 39,68083333 | 2,177738141  |
| 194  | 0     | 561       | B        | 3,8         | 18,59027778 | 39,68611111 | 2,182136629  |
| 186  | 0     | 509       | B        | 3,8         | 18,59833333 | 39,715      | 2,188473048  |
| 193  | 0     | 509       | B        | 3,8         | 18,55694444 | 39,72       | 2,172411527  |
| 151  | 1     | 509       | B        | 4,5         | 18,57361111 | 39,6875     | 2,189237075  |
| 152  | 1     | 509       | B        | 4,5         | 18,56611111 | 39,705      | 2,188943327  |
| 157  | 1     | 473       | B        | 4,9         | 18,58166667 | 39,70722222 | 2,192833723  |
| 159  | 1     | 455       | B        | 4,9         | 18,60416667 | 39,70055556 | 2,178025423  |
| 161  | 1     | 455       | B        | 4,9         | 18,58972222 | 39,67166667 | 2,174619856  |
| 164  | 1     | 455       | B        | 4,9         | 18,57361111 | 39,72138889 | 2,181220529  |
| 166  | 1     | 509       | B        | 4,5         | 18,58666667 | 39,70583333 | 2,191012815  |
| 167  | 1     | 455       | B        | 4,9         | 18,60888889 | 39,69305556 | 2,17351159   |
| 170  | 1     | 509       | B        | 4,6         | 18,58222222 | 39,68777778 | 2,189536326  |
| 171  | 1     | 509       | B        | 4,6         | 18,56555556 | 39,66944444 | 2,172625465  |
| 172  | 1     | 460       | B        | 4,6         | 18,56638889 | 39,70333333 | 2,189674315  |
| 175  | 1     | 539       | B        | 4,6         | 18,58916667 | 39,71833333 | 2,181614515  |
| 176  | 1     | 539       | B        | 4,9         | 18,58972222 | 39,68833333 | 2,186418264  |
| 177  | 1     | 500       | B        | 4,9         | 18,58916667 | 39,67833333 | 2,179895566  |
| 185  | 1     | 514       | B        | 4,5         | 18,58166667 | 39,69888889 | 2,196919841  |
| 184  | 1     | 561       | B        | 4,5         | 18,54944444 | 39,70055556 | 2,175924737  |
| 174  | 1     | 539       | B        | 5           | 18,565      | 39,73111111 | 2,171242757  |
| 192  | 1     | 533       | B        | 4,3         | 18,58277778 | 39,73083333 | 2,1733568    |
| 162  | 1     | 455       | B        | 4,9         | 18,5375     | 39,69138889 | 2,165221123  |
| 150  | 2     | 561       | B        | 4,6         | 18,58111111 | 39,66666667 | 2,189232387  |
| 158  | 2     | 455       | B        | 4,2         | 18,59777778 | 39,68611111 | 2,189232387  |
| 190  | 3     | 533       | B        | 4,3         | 18,53166667 | 39,71444444 | 2,187753071  |
| 187  | 3     | 510       | B        | 4,3         | 18,54833333 | 39,69055556 | 2,187753071  |
| 160  | 4     | 460       | B        | 4,6         | 18,61611111 | 39,75138889 | 2,184844323  |
| 169  | 4     | 494       | B        | 5           | 18,61444444 | 39,72222222 | 2,186826102  |
| 165  | 4     | 455       | B        | 4,9         | 18,59       | 39,73416667 | 2,18575233   |
| 153  | 5     | 505       | B        | 4,3         | 18,56611111 | 39,72638889 | 2,183205626  |
| 154  | 5     | 460       | B        | 4,5         | 18,54944444 | 39,74416667 | 2,193892028  |
| 155  | 5     | 561       | B        | 4,5         | 18,54944444 | 39,7575     | 2,187115582  |
| 156  | 5     | 455       | B        | 4,5         | 18,56611111 | 39,73722222 | 2,19024794   |
| 168  | 5     | 455       | B        | 4,9         | 18,57555556 | 39,74222222 | 2,184078796  |
| 173  | 5     | 445       | B        | 4,5         | 18,56638889 | 39,74666667 | 2,19155661   |
| 163  | 5     | 455       | B        | 4,9         | 18,52388889 | 39,75388889 | 2,171180199  |
| 178  | 6     | 561       | B        | 13,4        | 18,58166667 | 39,69222222 | 2,2          |
| 179  | 7     | 561       | B        | 8,3         | 18,57305556 | 39,75361111 | 2,2          |
| 180  | 8     | 500       | B        | 10,8        | 18,5725     | 39,72138889 | 2,2          |
| 181  | 9     | 500       | B        | 11,8        | 18,59888889 | 39,7325     | 2,2          |
| 182  | 10    | 500       | B        | 12,6        | 18,52277778 | 39,71638889 | 2,2          |
| 183  | 11    | 500       | B        | 13,3        | 18,53277778 | 39,69       | 2,2          |
| 188  | 12    | 539       | B        | 0,9         | 18,59833333 | 39,71416667 | 2,2          |
| 189  | 13    | 500       | M        | 0           | 18,615      | 39,69888889 | 2,2          |
| 191  | 14    | 445       | B        | 10,2        | 18,56611111 | 39,68166667 | 2,2          |
| 9999 | 0     | 541       | B        | 3,95        | 18,58711806 | 39,70711806 | 0            |
| 9999 | 1     | 498       | B        | 4,705263158 | 18,57805556 | 39,69950292 | 0            |
| 9999 | 2     | 508       | B        | 4,4         | 18,58944444 | 39,67638889 | 0            |
| 9999 | 3     | 521       | B        | 4,3         | 18,54       | 39,7025     | 0            |
| 9999 | 4     | 469       | B        | 4,833333333 | 18,60685185 | 39,73592593 | 0            |
| 9999 | 5     | 476       | B        | 4,585714286 | 18,55670635 | 39,74400794 | 0            |
| 9999 | 6     | 561       | B        | 13,4        | 18,58166667 | 39,69222222 | 0            |
| 9999 | 7     | 561       | B        | 8,3         | 18,57305556 | 39,75361111 | 0            |
| 9999 | 8     | 500       | B        | 10,8        | 18,5725     | 39,72138889 | 0            |
| 9999 | 9     | 500       | B        | 11,8        | 18,59888889 | 39,7325     | 0            |
| 9999 | 10    | 500       | B        | 12,6        | 18,52277778 | 39,71638889 | 0            |
| 9999 | 11    | 500       | B        | 13,3        | 18,53277778 | 39,69       | 0            |
| 9999 | 12    | 539       | B        | 0,9         | 18,59833333 | 39,71416667 | 0            |
| 9999 | 13    | 500       | M        | 0           | 18,615      | 39,69888889 | 0            |
| 9999 | 14    | 445       | B        | 10,2        | 18,56611111 | 39,68166667 | 0            |

Tabela 15 - Resultado do MST-Ci sobre o conjunto 3 de 50 plantios

| ID   | GRUPO | DENSIDADE | PRODUTIV | IDADE       | LATITUDE | LONGITUDE   | SIMILARIDADE |             |
|------|-------|-----------|----------|-------------|----------|-------------|--------------|-------------|
| 178  | 0     | 561       | B        |             | 13,4     | 18,58166667 | 39,69222222  | 2,2         |
| 179  | 1     | 561       | B        |             | 8,3      | 18,57305556 | 39,75361111  | 2,2         |
| 180  | 2     | 500       | B        |             | 10,8     | 18,5725     | 39,72138889  | 2,2         |
| 181  | 3     | 500       | B        |             | 11,8     | 18,59888889 | 39,7325      | 2,2         |
| 182  | 4     | 500       | B        |             | 12,6     | 18,52277778 | 39,71638889  | 2,2         |
| 183  | 5     | 500       | B        |             | 13,3     | 18,53277778 | 39,69        | 2,2         |
| 188  | 6     | 539       | B        |             | 0,9      | 18,59833333 | 39,71416667  | 2,2         |
| 189  | 7     | 500       | M        |             | 0        | 18,615      | 39,69888889  | 2,2         |
| 191  | 8     | 445       | B        |             | 10,2     | 18,56611111 | 39,68166667  | 2,2         |
| 193  | 9     | 509       | B        |             | 3,8      | 18,55694444 | 39,72        | 1,981267039 |
| 160  | 9     | 460       | B        |             | 4,6      | 18,61611111 | 39,75138889  | 2,152271993 |
| 159  | 9     | 455       | B        |             | 4,9      | 18,60416667 | 39,70055556  | 2,175726948 |
| 167  | 9     | 455       | B        |             | 4,9      | 18,60888889 | 39,69305556  | 2,169515637 |
| 154  | 9     | 460       | B        |             | 4,5      | 18,54944444 | 39,74416667  | 2,163274828 |
| 155  | 9     | 561       | B        |             | 4,5      | 18,54944444 | 39,7575      | 2,154026217 |
| 165  | 9     | 455       | B        |             | 4,9      | 18,59       | 39,73416667  | 2,177119107 |
| 153  | 9     | 505       | B        |             | 4,3      | 18,56611111 | 39,72638889  | 2,18365046  |
| 168  | 9     | 455       | B        |             | 4,9      | 18,57555556 | 39,74222222  | 2,173105458 |
| 156  | 9     | 455       | B        |             | 4,5      | 18,56611111 | 39,73722222  | 2,175561697 |
| 173  | 9     | 445       | B        |             | 4,5      | 18,56638889 | 39,74666667  | 2,168124322 |
| 164  | 9     | 455       | B        |             | 4,9      | 18,57361111 | 39,72138889  | 2,190229985 |
| 174  | 9     | 539       | B        |             | 5        | 18,565      | 39,73111111  | 2,179784308 |
| 152  | 9     | 509       | B        |             | 4,5      | 18,56611111 | 39,705       | 2,18983847  |
| 172  | 9     | 460       | B        |             | 4,6      | 18,56638889 | 39,70333333  | 2,18934976  |
| 186  | 9     | 509       | B        |             | 3,8      | 18,59833333 | 39,715       | 1,981613178 |
| 195  | 9     | 539       | B        |             | 3,8      | 18,59027778 | 39,72222222  | 1,984968707 |
| 196  | 9     | 561       | B        |             | 3,8      | 18,59027778 | 39,72083333  | 1,985723118 |
| 197  | 9     | 556       | M        |             | 4,3      | 18,59027778 | 39,71305556  | 1,98857961  |
| 158  | 9     | 455       | B        |             | 4,2      | 18,59777778 | 39,68611111  | 2,173196156 |
| 194  | 9     | 561       | B        |             | 3,8      | 18,59027778 | 39,68611111  | 1,976812049 |
| 199  | 9     | 539       | M        |             | 4,3      | 18,59027778 | 39,68083333  | 1,97283928  |
| 198  | 9     | 556       | M        |             | 4        | 18,59027778 | 39,69888889  | 1,985284362 |
| 151  | 9     | 509       | B        |             | 4,5      | 18,57361111 | 39,6875      | 2,180642894 |
| 170  | 9     | 509       | B        |             | 4,6      | 18,58222222 | 39,68777778  | 2,180579615 |
| 176  | 9     | 539       | B        |             | 4,9      | 18,58972222 | 39,68833333  | 2,178668796 |
| 150  | 9     | 561       | B        |             | 4,6      | 18,58111111 | 39,66666667  | 2,163169487 |
| 161  | 9     | 455       | B        |             | 4,9      | 18,58972222 | 39,67166667  | 2,165816461 |
| 177  | 9     | 500       | B        |             | 4,9      | 18,58916667 | 39,67833333  | 2,171256521 |
| 157  | 9     | 473       | B        |             | 4,9      | 18,58166667 | 39,70722222  | 2,195302089 |
| 166  | 9     | 509       | B        |             | 4,5      | 18,58666667 | 39,70583333  | 2,191044322 |
| 185  | 9     | 514       | B        |             | 4,5      | 18,58166667 | 39,69888889  | 2,189650836 |
| 175  | 9     | 539       | B        |             | 4,6      | 18,58916667 | 39,71833333  | 2,187716236 |
| 192  | 9     | 533       | B        |             | 4,3      | 18,58277778 | 39,73083333  | 2,182036191 |
| 171  | 9     | 509       | B        |             | 4,6      | 18,56555556 | 39,66944444  | 2,164349141 |
| 162  | 9     | 455       | B        |             | 4,9      | 18,5375     | 39,69138889  | 2,163195943 |
| 184  | 9     | 561       | B        |             | 4,5      | 18,54944444 | 39,70055556  | 2,175442405 |
| 187  | 9     | 510       | B        |             | 4,3      | 18,54833333 | 39,69055556  | 2,170753938 |
| 190  | 9     | 533       | B        |             | 4,3      | 18,53166667 | 39,71444444  | 2,161730646 |
| 169  | 9     | 494       | B        |             | 5        | 18,61444444 | 39,72222222  | 2,166936255 |
| 163  | 9     | 455       | B        |             | 4,9      | 18,52388889 | 39,75388889  | 2,14221795  |
| 9999 | 0     | 561       | B        |             | 13,4     | 18,58166667 | 39,69222222  | 0           |
| 9999 | 1     | 561       | B        |             | 8,3      | 18,57305556 | 39,75361111  | 0           |
| 9999 | 2     | 500       | B        |             | 10,8     | 18,5725     | 39,72138889  | 0           |
| 9999 | 3     | 500       | B        |             | 11,8     | 18,59888889 | 39,7325      | 0           |
| 9999 | 4     | 500       | B        |             | 12,6     | 18,52277778 | 39,71638889  | 0           |
| 9999 | 5     | 500       | B        |             | 13,3     | 18,53277778 | 39,69        | 0           |
| 9999 | 6     | 539       | B        |             | 0,9      | 18,59833333 | 39,71416667  | 0           |
| 9999 | 7     | 500       | M        |             | 0        | 18,615      | 39,69888889  | 0           |
| 9999 | 8     | 445       | B        |             | 10,2     | 18,56611111 | 39,68166667  | 0           |
| 9999 | 9     | 502       | B        | 4,512195122 |          | 18,5769851  | 39,710271    | 0           |





