SOBRAPO

PODes
Revista Eletrônica Pesquisa Operacional para o Desenvolvimento

Rio de Janeiro, v.7, n.1, p. 29-54, janeiro a abril de 2015

CONSTRUINDO TIPOLOGIAS DE CURVAS DE CARGA COM O PROGRAMA R

José Francisco Moreira Pessanha[a*], Vinícius Layter Xavier[b], Marcelo Rubens dos Santos do Amaral[a], Luiz da Costa Laurencel[a]

[a]*Universidade do Estado do Rio de Janeiro - UERJ, Rio de Janeiro-RJ, Brasil*
[b]*Universidade Federal do Rio de Janeiro - UFRJ, Rio de Janeiro-RJ, Brasil*

Resumo

Os perfis típicos de curvas de carga de consumidores e redes constituem informações fundamentais para a determinação das tarifas de uso dos sistemas de distribuição de energia elétrica. Destaca-se que a sinalização horária das tarifas é determinada em grande parte pelos perfis típicos da demanda por eletricidade. Neste artigo é apresentada uma implementação computacional em ambiente R dos métodos K-Means, K-Medoides e Ward, três métodos estatísticos multivariados para análise de agrupamentos, úteis na identificação de perfis típicos da demanda horária por eletricidade, uma etapa crítica do processo de revisão tarifária das distribuidoras de energia elétrica. O presente artigo contribui no sentido de fornecer uma alternativa eficaz e econômica para a construção das tipologias de curvas de carga.

Palavras-Chave: Análise de agrupamentos, K-Means, K-Medoides, Método de Ward, curvas de carga, tipologias

Abstract

Typical load profiles of consumers and networks are essential information for determining the use rates in electric power distribution systems. It is noteworthy that the time of use tariffs are based on the hourly load profiles. This paper presents a computational implementation in R environment of the K-Means, K-Medoids and Ward methods, three multivariate statistical methods for cluster analysis, useful in identifying typical daily load profiles, a critical stage in the revision of electricity tariff. This paper provides an effective and economical alternative to the construction of typologies of load curves.

Keywords: Cluster analysis, K-Means, K-Medoids, Ward Method, load curves, typical profiles

***Autor para correspondência:** e-mail: professorjfmp@hotmail.com

Versão inicial submetida em 12/08/2013. Versão final recebida em 28/11/2014.

1. Introdução

A forma de como a demanda por energia elétrica evolui ao longo do dia é uma informação fundamental na determinação das tarifas de eletricidade (BRASIL, 1985; PESSANHA et al, 2004, GUARDIA et al, 2010). Esta informação encontra-se na curva de carga diária das unidades consumidoras que descrevem trajetórias da demanda por energia elétrica ao longo das horas do dia. As formas assumidas pelas curvas de carga refletem os usos da energia elétrica pelos consumidores e apresentam perfis diferenciados nas distintas classes de consumo, conforme ilustrado na Figura 1.

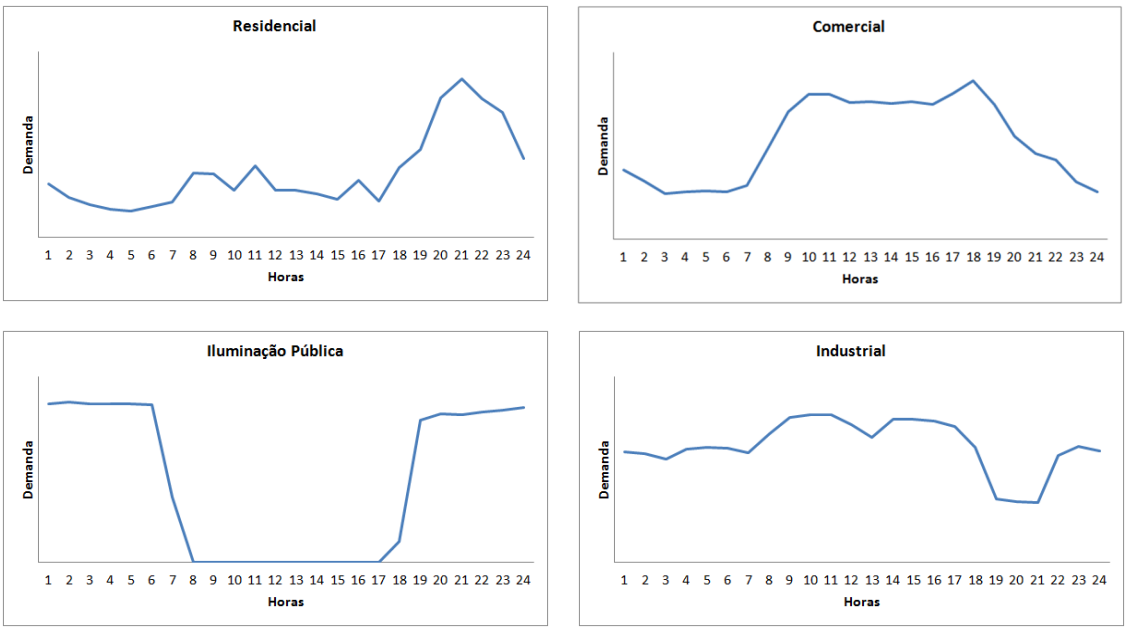

Figura 1 - Perfis típicos de curvas de carga de diferentes classes de consumo.

De forma bastante resumida, a obtenção dos perfis típicos de curvas de carga inicia-se com a coleta de medições de curvas de carga em uma amostra de clientes atendidos por uma concessionária de distribuição de energia elétrica (SCHROCK, 1997). No conjunto de curvas de carga amostradas é possível identificar alguns padrões, ou seja, curvas com perfis semelhantes entre clientes de uma mesma classe de consumo. Por exemplo, na classe residencial as tipologias de curvas de carga refletem hábitos de consumo que tem relação direta com a posse de eletrodomésticos e a rotina diária dos moradores das residências, formando padrões típicos do consumo residencial.

Assim, as curvas de carga diárias das unidades consumidoras amostradas podem ser classificadas em conjuntos mutuamente exclusivos ou *clusters*, de tal forma que os perfis diários

em um mesmo *cluster* sejam semelhantes entre si, porém diferentes dos perfis classificados nos demais *clusters*. Os agrupamentos não são definidos a priori, mas são identificados a partir dos dados por meio de algum método de análise de agrupamentos - *cluster analysis* (DIDAY, 1971; CHANTELOU et al, 1996; DEBRÉGEAS & HÉBRAIL, 1998; HÉBRAIL, 2001; PESSANHA et al 2002; GERBEC et al, 2004; FIGUEIREDO et al, 2005; CHICCO et al, 2006; SATHIRACHEEWIN, & SURAPATANA, 2011; RAMOS et al, 2012). A longa lista de referências acima indica que a análise de agrupamentos abrange uma variedade de métodos estatísticos multivariados que têm como propósito comum auxiliar a descoberta de uma estrutura latente em um conjunto de dados que permita classificá-los em grupos exaustivos e mutuamente exclusivos denominados *clusters*.

As formas dos perfis típicos correspondem aos perfis médios (centroides) dos agrupamentos. Ao final, os perfis médios são ajustados ao consumo anual do segmento de mercado que eles representam, obtendo-se os perfis típicos ou tipologias da curva de carga diária (PESSANHA et al, 2004). O processo de obtenção das tipologias é ilustrado na Figura 2.

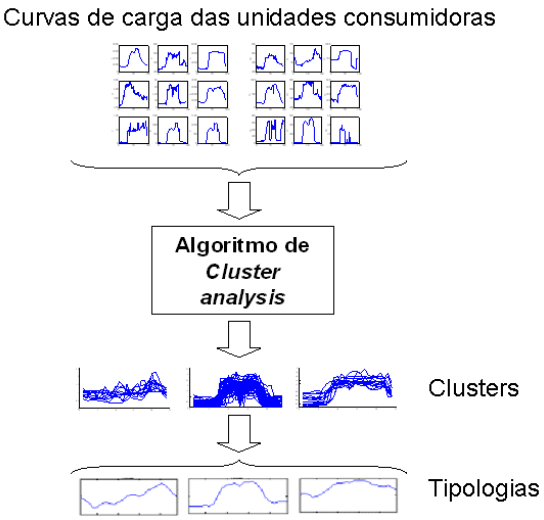

Figura 2 - Contrução das tipologias de carga (PESSANHA et al, 2006).

As tipologias são definidas para cada classe de consumo e representam a diversidade do comportamento dos consumidores da classe. Da mesma forma são definidas as tipologias de redes, que representam os perfis típicos das solicitações de potência em pontos selecionados da rede de distribuição (transformadores).

O objetivo deste artigo consiste em mostrar a construção de tipologias de curvas de carga por meio das funções de análise de agrupamentos disponibilizadas no programa R (R

CORE TEAM, 2013), um *software* livre e *open source*, altamente extensível, no qual são disponibilizadas várias funções para análise de dados e rotinas gráficas, nativas ou obtidas em pacotes (*packages*) distribuidos na internet. Adicionalmente, vale destacar que o *software* R é uma poderosa ferramenta de apoio ao ensino de métodos estatísticos (LEONI & COSTA, 2012).

Mais especificamente, neste artigo é apresentada uma implementação computacional em ambiente R dos métodos *K-Means*, K-Medoides e Ward, três métodos estatísticos multivariados para análise de agrupamentos (FREI, 2006; LATTIN et al, 2011).

A identificação de perfis típicos da demanda horária por eletricidade é uma etapa crítica do processo de revisão tarifária das distribuidoras de energia elétrica, pois requer a amostragem e a análise de um razoável número de medições de curva de carga. O presente artigo contribui no sentido de fornecer uma alternativa eficaz e econômica para a construção das tipologias de curvas de carga e que pode ser utilizada de forma complementar com aplicativos desenvolvidos para o cálculo tarifário (PESSANHA et al, 2004).

Para fins didáticos, a implementação computacional em ambiente R é ilustrada por meio de um exemplo no qual são consideradas medições de carga oriundas de um amostra de unidades consumidoras em baixa tensão. As curvas normalizadas estão disponíveis em https://www.academia.edu/9518147/curvas_de_carga.

Este artigo encontra-se organizado em oito seções. Na seção 2 tem-se uma introdução ao R. Na sequência, na seção 3 tem-se uma descrição dos procedimentos de importação e padronização dos dados, processos típicos em análise de dados. As implementações dos métodos de Ward e *K-Means* (EVERITT, 2007), métodos tradicionalmente utilizados no setor elétrico brasileiro (PESSANHA & LAURENCEL, 2009), são descritas nas seções 4 e 5 respectivamente. Já a implementação do método K-Medoides (KAUFMAN & ROUSSEEUW, 1986) é apresentada na seção 6. As medidas de validação (BROCK et al, 2008) dos três métodos supracitados são apresentadas na seção 7, tais medidas auxiliam na definição do número de agrupamentos. Por fim, na seção 8 são resumidas as principais conclusões do trabalho.

2. O programa R

O programa R é um *software* livre que pode ser obtido na página http://www.r-project.org/. Para instalar o R basta seguir os passos indicados abaixo:

I. Clique em *downloadR* ou *CRANmirror* na página do R.
II. Na lista de *CRAN mirror* escolha um repositório, por exemplo, a Fiocruz.
III. Escolha a plataforma Linux ou Windows.

IV. Escolha a opção *base*.

V. Execute o arquivo de instalação

O R já vem com um conjunto de funções que permitem realizar diferentes análises estatísticas, por exemplo, análise de regressão linear (função lm), análise de componentes principais (função princomp) e análise de agrupamentos (funções hclust e kmeans). O R também oferece diferentes formas de executar um programa, por exemplo, pode-se digitar o programa na própria janela do R. Adicionalmente, pode-se contar com editores de códigos como o RStudio (www.rstudio.org) e o Tinn-R (www.sciviews.org/Tinn-R).

3. Leitura e normalização dos dados

A título de exemplo, considere o arquivo *curva_de_carga.csv* localizado em um diretório *c:/exemplo* e contendo uma amostra de curvas de carga de unidades consumidoras selecionadas aleatoriamente entre os clientes de baixa tensão de uma concessionária de distribuição de energia elétrica. Trata-se de uma planilha na qual cada linha guarda a curva horária de uma unidade consumidora, enquanto cada coluna representa uma determinada hora do dia. A importação das curvas de carga para o R pode ser realizada por meio dos seguintes comandos (comentários após #):

```
setwd('c:/exemplo') # estabelece o diretório de trabalho
dados = read.csv('curva_de_carga.csv',sep=',',dec='.',header=FALSE) # importa os dados
head(dados) # visualiza dados
```

No comando *read.csv* acima, a opção *sep*=',' indica que as colunas estão separadas por vírgulas, a opção *dec*='.' estabelece que o separador decimal é o ponto, enquanto *header=FALSE* informa que o arquivo de dados não contém cabeçalho. O objeto *dados* armazena as curvas de carga, sendo que *dados*[*i,j*] guarda a demanda da hora *j* na curva de carga do cliente *i*. As seis primeiras curvas de carga podem ser visualizadas por meio do comando *head*(*dados*), conforme ilustrado na Figura 3.

```
> head(dados)
           [,1]        [,2]         [,3]       [,4]       [,5]       [,6]       [,7]       [,8]       [,9]      [,10]
[1,] 0.0380636108 0.045049371 0.0487720045 0.05075826 0.05326969 0.04565029 0.03923383 0.07986199 0.16850230 0.19042007
[2,] 0.0719037823 0.086530656 0.0975221868 0.10383425 0.10474408 0.08227005 0.06662111 0.08183957 0.10117736 0.08888612
[3,] 0.0878077474 0.075895844 0.0722750623 0.07337497 0.06275856 0.04085092 0.02682403 0.02766153 0.02854631 0.02469972
[4,] 0.1251753830 0.070064816 0.0530505201 0.08260494 0.11027828 0.07923830 0.03445217 0.07563106 0.12452899 0.12096366
[5,] 0.1093628755 0.082319809 0.0627050138 0.04300222 0.02501493 0.02603225 0.04567739 0.09987555 0.14081404 0.10522051
[6,] 0.0005633575 0.000488267 0.0001560271 0.00000000 0.00000000 0.00000000 0.02836798 0.07638774 0.08262543 0.04611361
          [,11]      [,12]      [,13]      [,14]      [,15]      [,16]     [,17]      [,18]     [,19]      [,20]     [,21]
[1,] 0.11580470 0.06861438 0.10434029 0.12720884 0.11728515 0.04461011 0.2125561 0.6391056 0.7761964 0.4801247 0.2503169
[2,] 0.08553554 0.08646031 0.08972446 0.08763256 0.08577737 0.07924557 0.1082485 0.1956863 0.2815024 0.2758444 0.2308338
[3,] 0.02884388 0.04860802 0.07520375 0.08624286 0.06528735 0.03291659 0.0929456 0.3574873 0.5121323 0.3791253 0.2045704
[4,] 0.11372105 0.13396909 0.12838744 0.11742970 0.11435404 0.12006637 0.1556027 0.2411081 0.4057570 0.4701564 0.3693418
[5,] 0.04948736 0.05942557 0.08764736 0.09619358 0.08306596 0.07338071 0.1941448 0.3616372 0.4012382 0.3299751 0.2741570
[6,] 0.02517921 0.02981123 0.03040450 0.02352645 0.01659287 0.02368003 0.0204946 0.0402676 0.1494280 0.2827739 0.3148272
          [,22]     [,23]      [,24]
[1,] 0.2058028 0.1673930 0.14130852
[2,] 0.2100023 0.1729726 0.10990457
[3,] 0.1583434 0.1152714 0.06696114
[4,] 0.2264596 0.1592020 0.15252028
[5,] 0.2131550 0.1724632 0.16044479
[6,] 0.2956449 0.2818607 0.26048135
```

Figura 3 - Seis primeiras curvas de carga contidas no arquivo *curva_de_carga.csv*.

Por meio do comando *dim(dados)* pode-se obter informações sobre as dimensões da matriz de dados (Figura 4). Neste caso há 74 curvas diárias descritas por 24 demandas horárias:

```
> dim(dados)
[1] 74 24
```

Figura 4 - Dimensões da matriz de dados.

O gráfico da primeira curva de carga da matriz de dados (Figura 5) pode ser obtido pelo seguinte comando, onde *seq(1,24,1)* gera a sequência de números inteiros entre 1 e 24:

```
plot(seq(1,24,1),dados[1,],type='l',xlab='horas',ylab='W')
```

Já o conjunto com todas curvas analisadas (Figura 6) pode ser visualizado por meio do seguinte comando:

```
matplot(matrix(seq(1,24,1),ncol=1),t(dados),type='l',ylab='kW',xlab='Horas')
```

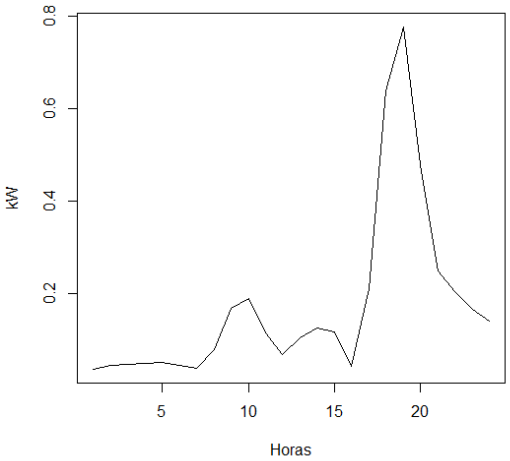

Figura 5 - Primeira curva de carga.

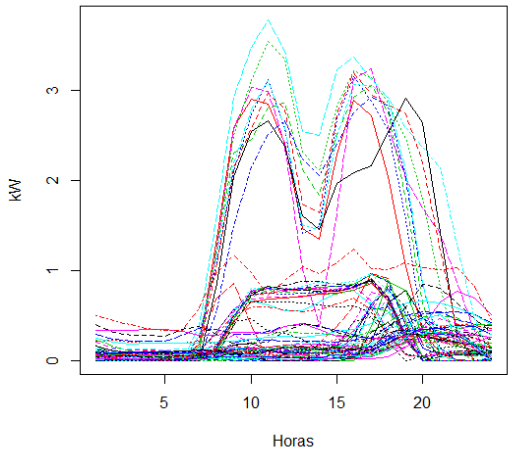

Figura 6 - Curvas de carga da matriz de dados.

SOBRAPO
PODes
Revista Eletrônica Pesquisa Operacional para o Desenvolvimento

Rio de Janeiro, v.7, n.1, p. 29-54, janeiro a abril de 2015

Dado que o objetivo consiste em identificar grupos de curvas com formas semelhantes, as medições devem ser normalizadas pelos respectivos valores médios, antes de serem processadas pela análise de agrupamentos (PESSANHA et al, 2004). O código em R para a obtenção das curvas normalizadas é indicado abaixo:

```
medias = apply(dados,1,mean) # calcula a demanda média em cada curva de carga
dadospu = sweep(dados,1,medias,FUN='/') # divide cada curva pela respectiva média
```

O comando abaixo produz o gráfico com todas as curvas normalizadas (Figura 7):

```
matplot(matrix(seq(1,24,1),ncol=1),t(dadospu),type='l',ylab='PU',xlab='horas')
```

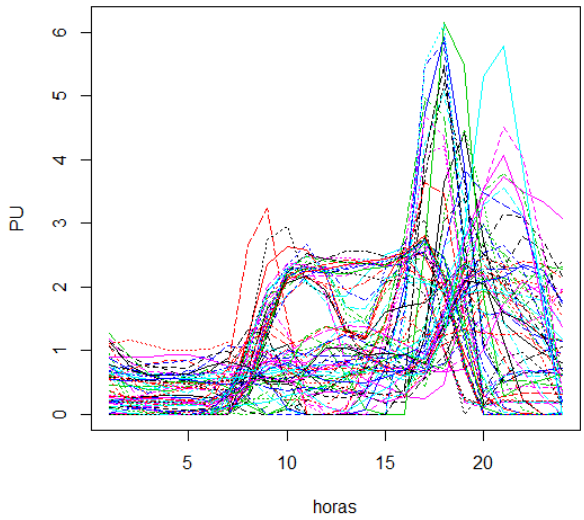

Figura 7 - Curvas normalizadas.

4. Método de WARD

O método de Ward é um método hierárquico aglomerativo, ou seja, inicialmente cada agrupamento (*cluster*) tem apenas um objeto (uma curva de carga), e em cada iteração de execução do algoritmo os *clusters* são agregados dois a dois até que reste apenas um grupo com todos os elementos.

Portanto o método de Ward agrupa sucessivamente os n grupos iniciais, em n-1, n-2,...,1 grupos, obtendo ao final uma estrutura em árvore semelhante à classificação biológica da taxonomia de Lineu que classifica os seres vivos em reino, filos, classes, ordens, famílias, gêneros e espécies. Esta estrutura de árvore é conhecida como dendrograma.

Os métodos hierárquicos baseiam-se em uma matriz simétrica de ordem n, onde o elemento ij guarda a distância entre os *clusters* i e j. Inicialmente as distâncias correspondem

Versão inicial submetida em 12/08/2013. Versão final recebida em 28/11/2014.

aos quadrados das distâncias euclidianas entre os próprios objetos, pois cada *cluster* tem apenas um elemento. Os *clusters* mais próximos são os mais semelhantes e, portanto, são os primeiros a serem agrupados. À medida que os *clusters* vão sendo agrupados, a ordem da matriz de distâncias diminui e as distâncias são recalculadas com base na seguinte fórmula:

$$d_{ij} = \frac{p_i p_j}{p_i + p_j} d^2\left(c_i, c_j\right) \qquad (1)$$

em que p_i e p_j denotam as quantidades de objetos nos *clusters* *i* e *j* respectivamente e $d^2(c_i,c_j)$ representa o quadrado da distância euclidiana entre os centroides dos agrupamentos *i* e *j*.

No R, a execução do método de Ward é realizada por meio do seguinte comando (EVERITT, 2007):

```
resultado.hc = hclust(dist(dadospu),method='ward',members=NULL)
```

A sequência de agregações é ilustrada no dendrograma (Figura 8), um gráfico útil na avaliação do número de agrupamentos em um conjunto de dados. O dendrograma é gerado pelo comando *plot(resultado.hc ,ylab='Distâncias',main='').*

O dendrograma oferece soluções para diferentes níveis de agregação dos objetos. Por exemplo, dois grandes ramos ou agrupamentos emergem do dendrograma ilustrado na Figura 8, sendo que eles são bem distintos, conforme sugerido pelos segmentos verticais que expressam o grau de dissimilaridade entre os agrupamentos. Quanto maior o comprimento destes segmentos verticais mais distintos são os ramos que convergem para um mesmo ponto no dendrograma, ou seja, mais distintos são os *clusters* agrupados em uma etapa do processo aglomerativo do método hierárquico.

Inicialmente são agrupados os objetos ou *clusters* mais semelhantes, portanto, na base do dendrograma os segmentos verticais são curtos. À medida que o processo de aglomeração se desenvolve, *clusters* cada vez mais distintos são agrupados e os segmentos verticais tornam-se cada vez mais longos. Assim, uma boa estratégia para a formação de agrupamentos homogêneos, consiste em observar o momento em que os ramos tornam-se longos e então classificar todos os objetos conectados ao ramo em um mesmo *cluster*. No caso da Figura 8 é evidente a presença de três ramos distintos, sendo que cada um deles origina um agrupamento. Portanto, o dendrograma ilustrado na Figura 8 sugere a presença de três agrupamentos (*k*=3). No ambiente R, a identificação das curvas classificadas em cada agrupamento pode ser realizada pelo seguinte comando:

```
clusters = cutree(resultado.hc,k=3)
```

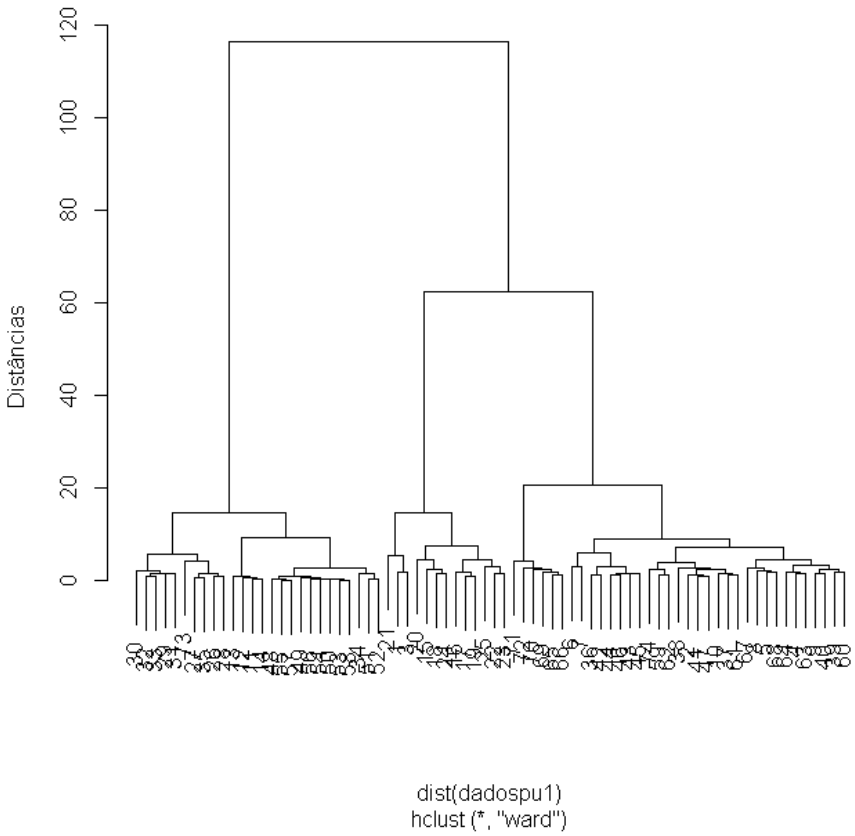

Figura 8 - Dendrograma, *plot(resultado.hc ,ylab='Distâncias',main='').*

Conforme ilustrado na Figura 9, o objeto *clusters* identifica o rótulo do agrupamento onde foi classificada cada linha da matriz de dados.

```
> clusters
 [1] 1 2 1 2 2 2 2 2 2 2 3 3 3 3 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 2 2 2 2
[40] 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2
```

Figura 9 - Identificação dos agrupamentos em que foram classificadas as curvas de carga.

Por exemplo, as curvas de carga nas linhas 1, 3, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 e 25 da matriz de dados foram classificadas no *cluster* 1. Para identificar estas linhas pode-se utilizar o comando *which(clusters==1)*. O número de curvas em cada agrupamento é obtido pelo comando *table*(*clusters*):

```
> table(clusters)
clusters
 1  2  3
13 35 26
```

Figura 10 - Número de curvas em cada agrupamento, *table*(*clusters*).

Uma representação gráfica das curvas classificadas em cada agrupamento é ilustrada na Figura 11, obtida pela seguinte sequência de comandos:

```
par(mfrow=c(3,1))
for (i in 1:3) {
cluster = which(clusters==i)
matplot(matrix(seq(1,24,1),ncol=1),t(dadospu[cluster,]),type='l',ylab='PU',xlab='horas',main=paste('cl
uster',i),cex.main=2,cex.axis=1.5) }
```

Ainda na Figura 11, vale ressaltar que os perfis classificados nos *clusters* 1 e 2 estão associados com unidades consumidoras da classe residencial, enquanto os perfis classificados no *cluster* 3 são típicos de unidades consumidoras que desenvolvem atividades comerciais e/ou industriais.

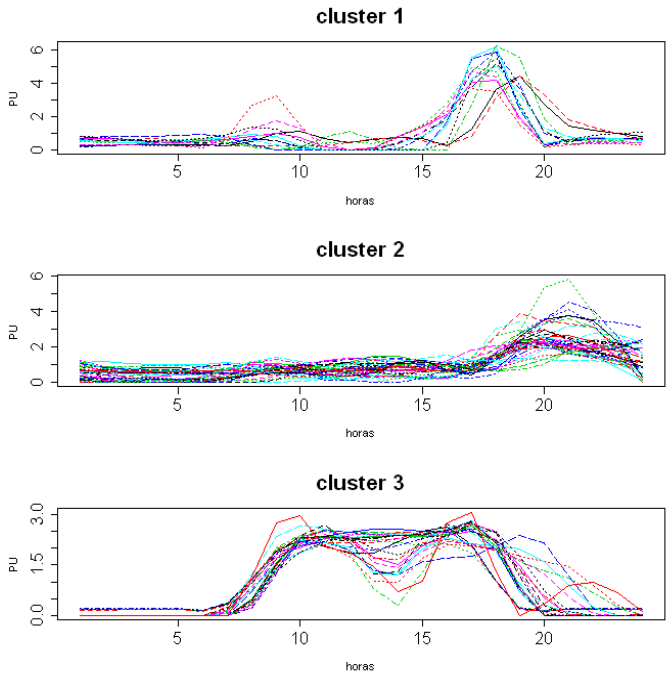

Figura 11 - Composição dos agrupamentos (Ward).

5. Método K-MEANS

A análise de agrupamentos também pode ser efetuada por meio do algoritmo *K-Means*, um método não-hierárquico. O *K-Means* classifica o conjunto de objetos em um número de categorias (*clusters*) especificado previamente. O número de *clusters* não é conhecido a priori, mas com base no dendrograma (LATTIN, et al, 2011) ilustrado na Figura 8 pode-se considerar que três *clusters* é uma boa solução inicial.

O critério utilizado pelo método *K-Means* na classificação dos objetos em *K clusters* consiste em minimizar a variabilidade dentro dos agrupamentos, expressa pela soma dos quadrados dos desvios entre as observações e o centroide do *cluster* no qual as observações foram alocadas (*within-group sum of squares* - WSS):

$$\text{WSS} = \sum_{k=1}^{K} \sum_{i \in cluster\_k} \sum_{j=1}^{p} \left(x_{ij} - c_{kj}\right)^2 \qquad (2)$$

onde x_{ij} é a demanda da i-ésima curva de carga na j-ésima variável (j=1,p), c_{kj} é a j-ésima coordenada do centroide do k-ésimo agrupamento e $i \in$ *cluster k* denota os índices de todos as curvas de carga classificadas no k-ésimo agrupamento.

Cada objeto pertence a apenas um dos *K clusters*, portanto, a classificação dos *n* objetos pode ser representada por uma matriz binária *U* de dimensões *n* x *k*, onde $u_{ij} = 1$ se o i-ésimo objeto pertence ao j-ésimo *cluster* e $u_{ij} = 0$ caso contrário. Se os centros dos *K clusters* são fixos, a partição ótima consiste em alocar cada objeto no *cluster* com o centroide mais próximo. Assim, os valores de u_{ij} são definidos pela seguinte regra:

$u_{ij} = 1$ se o centro de gravidade do j-ésimo agrupamento é o mais próximo do i-ésimo objeto.

$u_{ij} = 0$ caso contrário.

Por outro lado, para uma dada partição dos *n* objetos (matriz *U* fixa), o centroide do j-ésimo *cluster*, j=1,*K* é a média dos objetos classificados no *cluster*.

$$c_j = \frac{1}{n_j} \sum_{cluster\_k} x_i \qquad (3)$$

O método *K-Means* pode ser implementado segundo o algoritmo a seguir, onde a matriz *U* e os centroides dos *clusters* são obtidos de forma iterativa (JANG et al, 1997):

I. Inicialize os centroides dos *clusters* pelo sorteio de *K* objetos entre os *n* a serem classificados.
II. Determine a matriz *U* de acordo com o critério do centroide mais próximo, ou seja, aloque cada objeto ao *cluster* com o centroide mais próximo.
III. Calcule o valor da função objetivo WSS em (2). Pare se o valor da função estiver abaixo de uma tolerância pré-especificada, se a melhoria em relação à iteração anterior for desprezível ou se o número máximo de iterações for alcançado.
IV. Atualize os centroides dos *clusters* e volte para o passo 2.

Este algoritmo é computacionalmente eficiente e produz bons resultados se os *clusters* são compactos, esféricos e bem separados no espaço (JAIN et al, 2000). Entretanto, o algoritmo não garante a convergência para uma solução ótima e o seu desempenho depende dos centroides

iniciais escolhidos no passo 1. Por esta razão é recomendável executar o algoritmo diversas vezes com seleção aleatória dos centroides iniciais e ao final selecionar a melhor solução, ou seja, a que minimiza a WSS (LATTIN et al, 2011).

No R, a classificação das curvas normalizadas em três grupos, com 10 inicializações aleatórias dos centroides iniciais e no máximo 100 iterações do algoritmo *K-Means* pode ser realizada por meio do seguinte comando (EVERITT, 2007):

```
resultado.kmeans = kmeans(dadospu,centers=3,iter.max=100,nstart=10)
```

Diferentes implementações computacionais do algoritmo descrito acima resultam em uma variedade de algoritmos para o *K-Means*, por exemplo, o algoritmo de Hartigan e Wong (1979) é o *default* disponibilizado pela função *kmeans*. A lista contendo os rótulos dos agrupamentos em que foram classificadas as curvas de carga (*Clustering vector*) é obtida pelo comando *resultado.kmeans$cluster*, conforme a Figura 12. O número de curvas de carga em cada *cluster* é obtido pelo comando *resultado.kmeans$size*.

```
> resultado.kmeans$cluster
 [1] 1 1 1 1 1 1 1 1 1 1 3 3 3 3 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 1 1 1
[39] 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 3 1
```

Figura 12 - Classificação das curvas de carga, *resultado.kmeans$cluster.*

```
> resultado.kmeans$size
[1] 37 11 26
```

Figura 13 - Número de curvas em cada agrupamento, *resultado.kmeans$size.*

A visualização das curvas classificadas em cada agrupamento (Figura 14) pode ser alcançada pela seguinte sequência de comandos:

```
par(mfrow=c(3,1))
for (i in 1:3) {
cluster = which(resultado.kmeans$cluster ==i)
matplot(matrix(seq(1,24,1),ncol=1),t(dadospu[cluster,]),type='l',ylab='PU',xlab='horas',main=paste('cl
uster',i),cex.main=2,cex.axis=1.5)
}
```

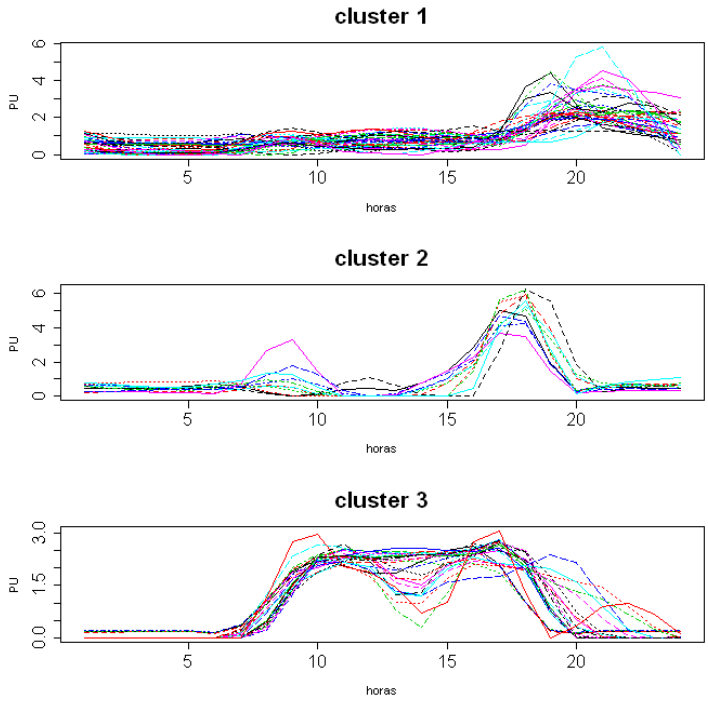

Figura 14 - Composição dos agrupamentos (*K-Means*).

As coordenadas dos centroides dos agrupamentos na Figura 15 podem ser extraídas pelo comando *resultado.kmeans$centers*. Os centroides definem as formas das tipologias das curvas de carga dos clientes, uma informação fundamental para o cálculo das tarifas de uso dos sistemas de distrribuição de energia elétrica (PESSANHA et al, 2004).

```
> resultado.kmeans$centers
        [,1]       [,2]       [,3]       [,4]       [,5]       [,6]       [,7]
1 0.60120123 0.45141491 0.38049250 0.37232099 0.36743777 0.36897194 0.45523675
2 0.51575603 0.49327127 0.46812654 0.43853038 0.46102117 0.50184295 0.58735561
3 0.02908811 0.02930518 0.02944747 0.02972227 0.02872947 0.02187252 0.09858903
       [,8]      [,9]      [,10]      [,11]      [,12]      [,13]      [,14]      [,15]
1 0.5985181 0.6834273 0.6840149 0.6663031 0.7029666 0.7568194 0.7785687 0.7583134
2 0.8536413 0.8580840 0.4346498 0.1482565 0.1412942 0.1080174 0.3292199 0.7151407
3 0.7139732 1.6750991 2.2291822 2.3211697 2.2065320 1.9175194 1.8658763 2.1510887
      [,16]      [,17]     [,18]     [,19]      [,20]      [,21]      [,22]      [,23]
1 0.7252732 0.8911334 1.539173 2.244802 2.5062881 2.4162582 2.1630240 1.73819986
2 1.7318334 4.3724492 5.176492 2.854942 0.6546008 0.5108250 0.5986176 0.61094283
3 2.4209524 2.4635172 1.935002 1.026358 0.4393279 0.2749206 0.1495164 0.07818307
       [,24]
1 1.16737000
2 0.61797812
3 0.03898857
```

Figura 15 - Centroides dos agrupamentos, *resultado.kmeans$centers.*

A visualização dos centroides na Figura 16 pode ser gerada pela seguinte sequência de comandos:

```
par(mfrow=c(3,1))
for (i in 1:3) {
matplot(matrix(seq(1,24,1),ncol=1),
resultado.kmeans$centers[i,],type='l',ylab='PU',xlab='horas',main=paste('cluster',i),cex.main=2,cex.axis=1.5) }
```

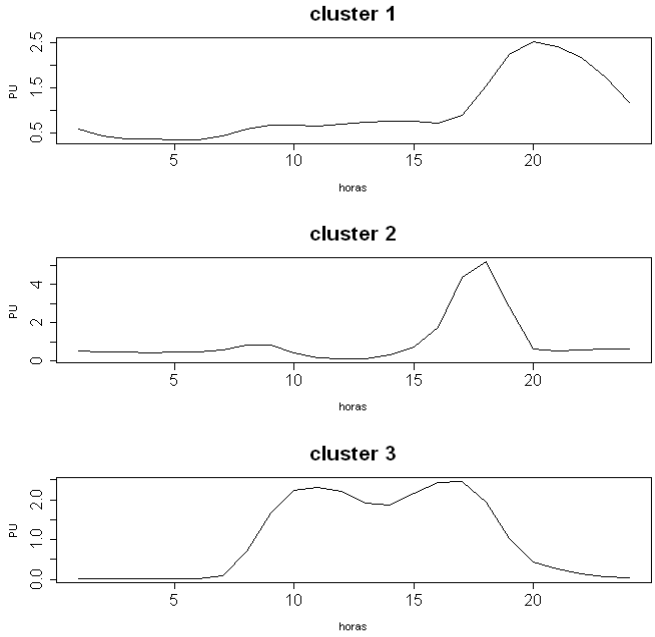

Figura 16 - Perfis típicos das curvas de carga.

A variabilidade total do conjunto de dados analisados (74 objetos descritos por 24 variáveis) é quantificada pela soma dos quadrados totais (TSS – *total sum of squares*) ou inércia total definida pela soma dos quadrados dos desvios dos objetos em relação ao centroide do conjunto de dados ($\bar{x}$):

$$\text{Inércia total} = \sum_{i=1}^{n}\sum_{j=1}^{p}\left(x_{ij} - \bar{x}_j\right)^2 \qquad (4)$$

em que *n* é o total de objetos (curvas de carga) a serem classificados, *p* é o número de variáveis que descrevem os objetos, x_{ij} é a demanda da i-ésima curva de carga na j-ésima hora e $\bar{x}_j$ é a demanda média na hora *j*.

A inércia total pode ser decomposta em duas parcelas: a inércia entre os agrupamentos (BSS – *between sum of squares*) e a inércia dentro dos agrupamentos (WSS):

$$Inércia\ total\ =\ TSS\ =\ BSS\ +\ WSS \qquad (5)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{p}\left(x_{ij} - \bar{x}_j\right)^2 = \sum_{k=1}^{K} n_k \sum_{j=1}^{p}\left(c_{kj} - \bar{x}_j\right)^2 + \sum_{k=1}^{K}\sum_{i \in cluster\_k}\sum_{j=1}^{p}\left(x_{ij} - c_{kj}\right)^2 \qquad (6)$$

em que n_k é o número de objetos classificados no k-ésimo *cluster*, c_{kj} é o j-ésimo elemento do centroide do k-ésimo agrupamento e $i \in cluster\_k$ denota os índices de todos as curvas de carga classificadas no k-ésimo agrupamento.

A inércia entre os agrupamentos (BSS) expressa a variabilidade entre os centroides dos agrupamentos. A BSS é a soma dos quadrados dos desvios entre os centroides dos agrupamentos e a curva média, ponderados pelos totais de elementos classificados em cada *cluster*:

$$\text{BSS} = \sum_{k=1}^{K} n_k \sum_{j=1}^{p} \left(c_{kj} - \bar{x}_j\right)^2 \qquad (7)$$

Já a inércia dentro dos agrupamentos (WSS) expressa a heterogeneidade interna dos *clusters*. Cada parcela da WSS corresponde à um *cluster* e é definida pela soma dos quadrados dos desvios entre o centroide do *cluster* e os objetos que nele foram classificados.

$$\text{WSS} = \sum_{k=1}^{K} WSS_k = \sum_{k=1}^{K} \sum_{i \in cluster\_k} \sum_{j=1}^{p} \left(x_{ij} - c_{kj}\right)^2 \qquad (8)$$

onde WSS_k corresponde a contribuição da variabilidade dentro do cluster k para a WSS.

Neste exemplo, a inércia total, obtida pelo comando *resultado.kmeans$totss*, é 1218,097, sendo que 866,7107 (*resultado.kmeans$betweenss*) deve-se a variabilidade entre os centróides dos agrupamentos, enquanto, 351,3866 (*resultado.kmeans$tot.withinss*) é devido a variabilidade dentro dos *clusters*.

A soma dos quadrados dentro de cada *cluster* (WSS) é obtida pelo comando *resultado.kmeans$withinss***.** Conforme ilustrado na Figura 17, o *cluster* 3 tem a maior variabilidade interna**.**

```
> resultado.kmeans$withinss
[1]  66.58334  68.13875 216.66456
```

Figura 17 - Inércia dentro dos agrupamentos, *resultado.kmeans$withinss.*

Ao classificar objetos busca-se formar *clusters* que tenham grande homogeneidade interna e que sejam diferentes dos demais agrupamentos. Ou seja, busca-se uma classificação em que a maior parte da variabilidade esteja entre os *clusters* e não dentro deles. Neste caso, verificou-se que com apenas três agrupamentos a maior parte da variabilidade (71%) reside entre os agrupamentos. Contudo, a inércia dentro dos agrupamentos (WSS) é uma medida problemática, pois diminui inexoravelmente à medida que o número de agrupamentos aumenta (LATTIN et al, 2011). Uma estatística mais apropriada para este fim é o pseudo-F, definido

pela razão do quadrado médio entre os *clusters BSS*/(*k*-1) e o quadrado médio dentro dos *clusters WSS*/(*N*-*k*):

$$pseudo - F = \frac{BSS/(k-1)}{WSS/(N-k)} \quad (9)$$

Por meio da estatística pseudo-F pode-se levar em conta os *tradeoffs* entre o aumento da homogeneidade interna dos agrupamentos e o número de *clusters*. O pseudo-F deve ser calculado para diferentes níveis de agregação (*k*), sendo que o número adequado de *clusters* corresponde ao maior valor da estatística (LATTIN et al, 2011). O código a seguir obtém a participação da inércia entre os agrupamentos (%BSS) na inércia total e o pseudo-F para níveis de agregação entre 2 e 9 *clusters*:

```
pseudoF = numeric(0)
bss = numeric(0)
nobs=dim(dadospu)[1] # número de curvas de carga
for (i in 2:9) {
    resultado.kmeans = kmeans(dadospu,i,nstart=10)
    auxpseudoF = (resultado.kmeans$betweenss/(i-1))/(resultado.kmeans$tot.withinss/(nobs-i))
    auxbss = resultado.kmeans$betweenss/resultado.kmeans$totss
    pseudoF = c(pseudoF,auxpseudoF)
    bss = c(bss,auxbss)
}
par(mfrow=c(1,2))
plot(seq(2,9,1),pseudoF,xlab= 'número de clusters ',ylab= 'pseudoF ',pch=1)
plot(seq(2,9,1),bss*100,xlab= 'número de clusters ',ylab= '%BSS ',pch=1)
```

Na Figura 18, pode-se observar que o pseudo-F é máximo na solução com três agrupamentos, sendo que neste caso a inércia entre os agrupamentos corresponde a cerca de 71% da inércia total, ou seja, a maior parte da variabilidade reside entre os agrupamentos e não dentro deles e, portanto, é uma solução satisfatória.

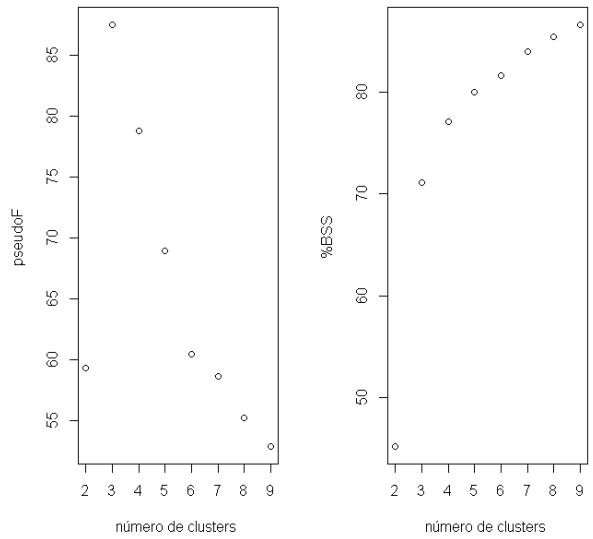

Figura 18 - Estatística Pseudo-F e participação da inércia entre *clusters* (%BSS).

6. Método k-MEDOIDES

No método K-Medoides os agrupamentos são definidos como subconjuntos de pontos que estão mais próximos dos seus respectivos elementos representativos, denominados medoides. O medoide de um determinado agrupamento pode ser definido como o objeto do grupo, cuja soma das distâncias a todos os demais objetos do mesmo grupo seja mínima.

Assim, o critério utilizado é a mínima soma de distância das observações ao medoide do grupo que ela pertence ao invés de mínima soma dos quadrados das distâncias das observações ao centro de gravidade do grupo, conforme no *K-Means*. Portanto, a diferença básica do K-Medoides em relação ao *K-Means* reside na utilização de uma das observações do conjunto original como elemento representativo, localizada mais ao centro do *cluster,* ao invés da escolha do centro de gravidade do grupo. Entre os algoritmos da família dos métodos K-Medoides destaca-se o algoritmo *Partitioning Around Medoides* - PAM (KAUFMAN & ROUSSEEUW, 1986), descrito de forma sucinta nos cinco passos a seguir:

I. Selecione aleatoriamente *k* medoides iniciais entre as *m* observações do conjunto de dados.

II. Associe cada observação do conjunto de dados ao medoide mais próximo.

III. Para cada medoide *m*:

Para cada não-medoide *q*:

Torne *q* um medoide e *m* um não medoide e calcule o custo total da configuração, ou seja, a dissimilaridade média de *q* em relação aos elementos associados ao medoide *m*. Selecione o medoide *q* com o menor custo.

IV. Repita os passos II - III até que não haja mudança nos medoides.

Por meio da função *pam* disponível no pacote *cluster* do R pode-se classificar as curvas normalizadas em três grupos:

```
library('cluster')  #  carrega a biblioteca cluster
resultado.pam = pam(x=dadospu,k=3) # classifica as curvas normalizadas em três grupos
```

Os índices que identificam as curvas qualificadas como medoides são obtidos pelo comando *resultado.pam$id.med* e neste exemplo correspondem aos objetos 47, 54 e 17, cujas coordenadas são os respectivos valores normalizados (*resultado.pam$medoids*) e definem as tipologias ilustradas na Figura 19, gerada pelo código abaixo:

```
par(mfrow=c(3,1))
for (i in 1:3) {
matplot(matrix(seq(1,24,1),ncol=1),
resultado.pam$medoids[i,],type='l',ylab='PU',xlab='horas',main=paste('cluster',i),cex.main=2,cex.axis
=1.5) }
```

Os índices que identificam as observações classificadas em cada agrupamento são disponibilizados pelo comando *resultado.pam$clustering*. Assim, pode-se visualizar a composição de cada *cluster*, conforme ilustrado na Figura 20, gerada pelo seguinte código:

```
par(mfrow=c(3,1))
for (i in 1:3) {
cluster = which(resultado.pam$clustering ==i)
matplot(matrix(seq(1,24,1),ncol=1),t(dadospu[cluster,]),type='l',ylab='PU',xlab='horas',main=paste('cl
uster',i),cex.main=2,cex.axis=1.5) }
```

Em relação ao tempo de processamento, o PAM é menos eficiente do que o *K-Means*, pois o cálculo do medoide é mais custoso computacionalmente do que o cálculo do centro de gravidade, resultando num maior tempo de processamento.

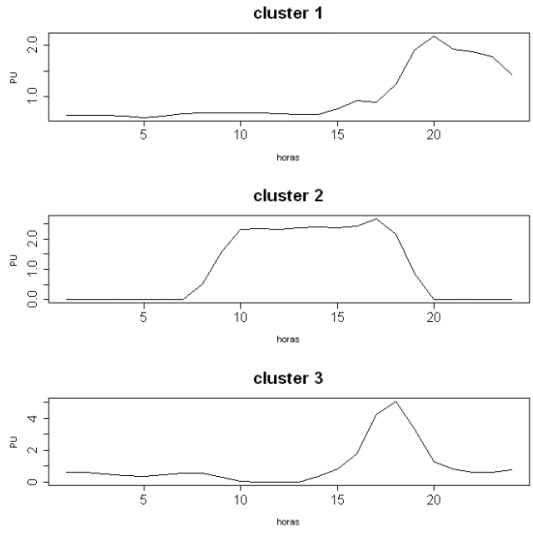

Figura 19 – Medoides.

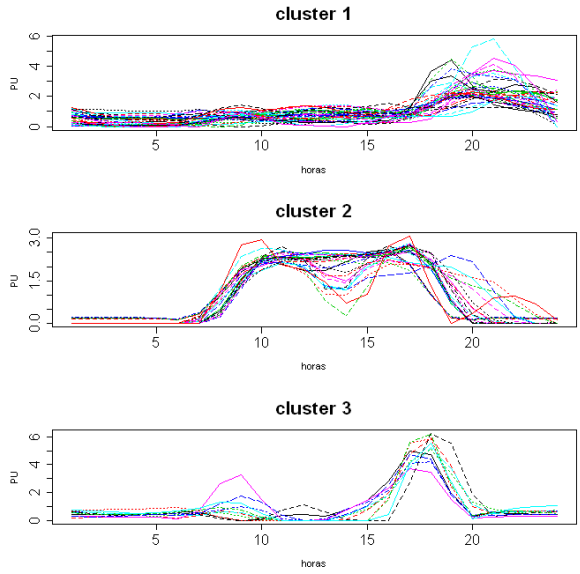

Figura 20 - Composição dos agrupamentos (K-Medoides).

7. Medidas de validação

Independentemente do método de análise de agrupamentos utilizado, os *clusters* resultantes devem ser compactos, bem separados e estáveis. No método de Ward a qualidade dos agrupamentos pode ser percebida por meio do dendrograma, enquanto no método *K-Means* as inércias intra (WSS) e inter (BSS) *cluster* fornecem medidas do grau de compacidade dos agrupamentos e da separação entre eles respectivamente.

Contudo, há medidas mais gerais, denominadas por medidas de validação, que permitem avaliar a qualidade dos agrupamentos identificados, por exemplo, medidas de conectividade

(relacionadas com o grau de vizinhança entre objetos classificados em um mesmo *cluster*), silhueta (homogeneidade interna) e índice Dunn (separação entre os agrupamentos) e medidas da estabilidade dos agrupamentos. Tais medidas podem ser aplicadas na validação dos resultados de qualquer método de análise de agrupamento e são úteis na definição do número de agrupamentos. No ambiente R, o pacote *clValid* (BROCK et al, 2008) disponibiliza funções que permitem calcular as medidas de validação. Adicionalmente, vale destacar que o mesmo pacote disponibiliza diversos algoritmos para análise de agrupamentos, incluindo os métodos de Ward, *K-Means*, *self-organizing maps* (SOM), *Unweighted Pair Group Method with Arithmetic Mean* (UPGMA), PAM, Diana, Fann e *Self-organizing tree algorithm*.

Considere um conjunto com N elementos descritos por p variáveis e classificados em K *clusters* $C_1,...,C_k$. Seja L o número de vizinhos de uma observação x_i, por exemplo, se L=1, apenas o elemento mais próximo de x_i é considerado seu vizinho. Denotando os L vizinhos de x_i por $nn_{i,j}$, j=1,L, define-se a seguinte variável $z_{i,j}$:

$z_{i,j} = 0$, se x_i e $nn_{i,j}$ pertencem ao mesmo *cluster*

$z_{i,j} = 1/j$, se x_i e $nn_{i,j}$ não pertencem ao mesmo *cluster*

Assim, a medida de conectividade para uma solução com K *clusters* é dada por:

$$Conectividade = \sum_{i=1}^{N} \sum_{j=1}^{L} z_{i,j} \qquad (10)$$

A conectividade varia entre zero e infinito e na partição ideal em K agrupamentos deve ser minimizada.

O comprimento da silhueta é a média das silhuetas das observações. A silhueta de uma observação x_i permite avaliar se a mesma foi bem classificada entre os K agrupamentos possíveis. A silhueta de uma observação x_i classificada em um *cluster k* é calculada da seguinte forma:

$$S_i = \frac{b_i - a_i}{\max(b_i, a_i)} \qquad (11)$$

em que a_i é a média das distâncias entre x_i e as observações classificadas no mesmo *cluster k* e b_i é a média das distâncias entre x_i e as observações no agrupamento vizinho mais próximo do *cluster k*.

A silhueta de uma observação assume valores no intervalo [-1,+1], sendo desejável um valor próximo de +1, indicando que a observação x_i está mais próxima das observações do *cluster* em que ela foi alocada e não do *cluster* vizinho. Portanto, uma boa partição dos objetos em K *clusters* deve maximizar a silhueta.

No R, a função *silhouette* disponível no pacote *cluster* gera o gráfico da silhueta dos elementos, por exemplo, para o caso da agregação das curvas de carga em 2, 3 e 4 agrupamentos pelo método PAM, os seguintes comandos produzem a Figura 21, na qual é possível visualizar as silhuetas das observações e a silhueta média em cada agrupamento.

```
par(mfrow=c(1,3))
plot(silhouette(pam(dadospu,2)))
plot(silhouette(pam(dadospu,3)))
plot(silhouette(pam(dadospu,4)))
```

Na Figura 21, a máxima silhueta média é 0,55 e ocorre na agregação em três *clusters*. Ainda na Figura 21, nota-se que as silhuetas das observações são dispostas em ordem decrescente dentro de cada agrupamento. No caso da solução em dois grupos, o primeiro grupo apresenta observações com silhuetas menores e com valores negativos em alguns casos, o que reduz a silhueta média, indicando que não se trata de uma boa solução. Na solução com quatro agrupamentos, os dois primeiros grupos apresentam silhuetas reduzidas contribuindo para a redução da silhueta média.

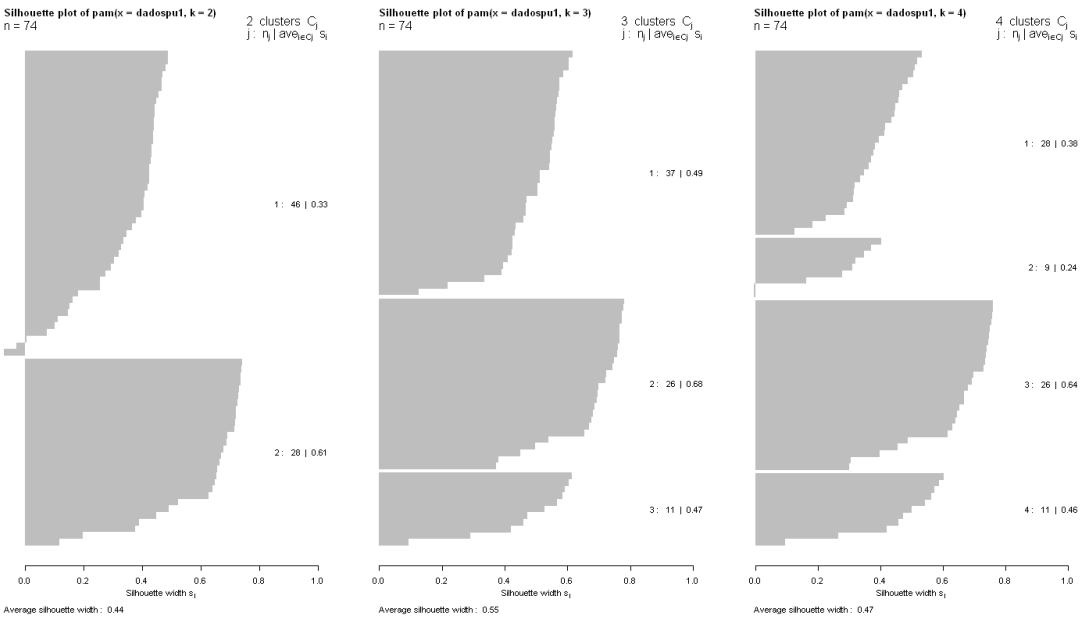

Figura 21 - Silhuetas das observações para agregação em 2, 3 e 4 agrupamentos.

O índice Dunn é a razão da menor distância entre duas observações em *clusters* distintos pela maior distância entre dois *clusters*. O índice Dunn assume valores no intervalo [0,1] e visando obter agrupamentos bem separados, o índice Dunn deve ser maximizado.

A seguir, na Figura 22, tem-se o resultado da validação da análise de agrupamentos na qual foram comparados os três métodos descritos ao longo deste artigo para diferentes níveis de agregação entre 2 e 9 agrupamentos. Os resultados apresentados na Figura 22 foram obtidos por meio da seguinte sequência de comandos:

```
library(clValid) # carrega o pacote clValid
intern=clValid(dadospu,2:9,clMethods=c("hierarchical","kmeans","pam"),validation="internal")
summary(intern)
```

Além das medidas de validação, o pacote *clValid* disponibiliza medidas que permitem avaliar a estabilidade dos agrupamentos. Tais medidas são calculadas a partir da comparação das partições geradas pela análise de agrupamentos do conjunto completo de dados (*p* variáveis) e pela análise de agrupamentos aplicada em conjuntos de dados incompletos, i.e., sem uma das variáveis (*p*-1 variáveis), nos quais uma coluna é removida por vez. Mais especificamente, o pacote *clValid* disponibiliza quatro estatísticas que permitem avaliar a estabilidade dos *clusters* resultantes (BROCK et al, 2008):

- APN - *average proportion of non-overlap*: Proporção média de observações não classificadas no mesmo *cluster* nos casos com dados completos e incompletos. O valor da estatística APN pertence ao intervalo [0,1], sendo que valores próximos de zero indicam agrupamentos consistentes.
- AD - *average distance*: Distância média entre observações classificadas no mesmo *cluster* nos casos com dados completos e incompletos. A estatística AD assume valores não negativos, sendo preferíveis valores próximos de zero.
- ADM - *average distance between means*: Distância média entre os centroides quando as observações estão em um mesmo *cluster*. A estatística ADM também assume valores não negativos, sendo preferíveis valores próximos de zero.
- FOM - *figure of merit*: A figura de mérito é uma medida do erro cometido ao usar os centroides como estimativas das observações na coluna removida. A estatística FOM assume valores não negativos, sendo preferíveis valores próximos de zero.

Os resultados da análise de estabilidade dos agrupamentos para diferetentes níveis de agregação são apresentados na Figura 23, obtida pelos seguintes comandos:

```
estabilidade=clValid(dadospu,2:9,clMethods=c("hierarchical","kmeans","pam"),validation="stability"
)
summary(estabilidade)
```

```
> summary(intern)

Clustering Methods:
 hierarchical kmeans pam

Cluster sizes:
 2 3 4 5 6 7 8 9

Validation Measures:
                             2      3      4      5      6      7      8      9

hierarchical Connectivity   1.8317  2.0429  5.1107  8.1913 17.1262 20.9214 22.1516 25.1917
             Dunn           0.3679  0.3576  0.3576  0.3486  0.3974  0.4429  0.4565  0.4565
             Silhouette     0.3535  0.5531  0.5233  0.4924  0.4285  0.4265  0.4151  0.3368
kmeans       Connectivity   1.8317  2.0429  9.0726 16.7853 26.1714 28.8060 30.0361 38.2671
             Dunn           0.3679  0.3576  0.2933  0.2574  0.3630  0.3473  0.4050  0.2422
             Silhouette     0.3535  0.5531  0.4930  0.4563  0.4431  0.4167  0.4125  0.3545
pam          Connectivity   6.6730  2.0429 12.7567 18.6988 23.5214 31.3341 33.0155 48.5357
             Dunn           0.1488  0.3576  0.2954  0.1444  0.1444  0.1856  0.1856  0.1631
             Silhouette     0.4380  0.5531  0.4695  0.3838  0.4075  0.3811  0.3787  0.3152

Optimal Scores:

             Score  Method       Clusters
Connectivity 1.8317 hierarchical 2
Dunn         0.4565 hierarchical 8
Silhouette   0.5531 hierarchical 3
```

Figura 22 - Medidas de validação dos agrupamentos.

```
> summary(estabilidade)

Clustering Methods:
 hierarchical kmeans pam

Cluster sizes:
 2 3 4 5 6 7 8 9

Validation Measures:
                       2      3      4      5      6      7      8      9

hierarchical APN  0.0344 0.0032 0.0063 0.0254 0.0174 0.0222 0.0170 0.0228
              AD   4.4763 2.7429 2.6534 2.5722 2.3180 2.1821 2.0799 2.0330
              ADM  0.2597 0.0249 0.0377 0.1061 0.1390 0.1250 0.0771 0.1106
              FOM  0.7192 0.4351 0.4129 0.4082 0.3847 0.3629 0.3486 0.3480
kmeans        APN  0.0379 0.0000 0.0124 0.0430 0.0112 0.0346 0.0238 0.0370
              AD   4.4796 2.7339 2.6143 2.5263 2.2079 2.1432 2.0443 1.9102
              ADM  0.2638 0.0000 0.0636 0.1786 0.0545 0.1255 0.0856 0.1966
              FOM  0.7192 0.4173 0.4126 0.3927 0.3576 0.3575 0.3499 0.3402
pam           APN  0.0152 0.0011 0.0166 0.0228 0.0311 0.0454 0.0560 0.0268
              AD   3.6720 2.7366 2.4589 2.2651 2.1572 2.0556 1.9347 1.8007
              ADM  0.1209 0.0079 0.0756 0.1070 0.1499 0.1932 0.1818 0.0952
              FOM  0.5553 0.4225 0.3934 0.3891 0.3838 0.3689 0.3573 0.3392

Optimal Scores:

    Score  Method Clusters
APN 0.0000 kmeans 3
AD  1.8007 pam    9
ADM 0.0000 kmeans 3
FOM 0.3392 pam    9
```

Figura 23 - Medidas de estabilidade dos agrupamentos.

Os resultados da análise de validação na Figura 22 indicam que o método de Ward foi o melhor nos três critérios, embora não exista uma concordância quanto ao número de *clusters*. Contudo, a medida de conectividade e a silhueta média recomendam um reduzido número de agrupamentos (2 ou 3) em oposição ao índice Dunn, o qual recomenda oito agrupamentos. Adicionalmente, vale notar que na partição em três agrupamentos os métodos de Ward, *K-Means* e PAM apresentam os mesmos valores para as estatísticas de validação, conforme ilustrado na Figura 22.

No que tange a estabilidade dos agrupamentos, as estatísticas apresentadas na Figura 23 também não exibem um consenso. Porém, vale notar que na solução com três agrupamentos proposta pelo *K-Means* as estatísticas APN e ADM atingem o mínimo permitido, ou seja, são valores ótimos. Já nas estatísticas AD e FOM, as discrepâncias da solução com três agrupamentos em relação aos respectivos valores ótimos alcançados não são relativamente tão expressivas (menores que dois desvios padrão).

Portanto, os resultados da análise de validação e estabilidade indicam que a melhor alternativa consiste em dividir o conjunto de curvas de carga em três agrupamentos. A decisão da partição com três *clusters* é reforçada pelo dendrograma (Figura 8), pelo valor elevado do pseudo-F (Figura 18) e pela participação expressiva da inércia entre os agrupamentos na inércia total (BSS concentra cerca de 71% da inércia total). Ressalta-se que os três métodos resultam em agrupamentos e tipologias semelhantes, conforme ilustrado nas Figuras 11, 14 e 20, nas quais se observam dois agrupamentos contendo perfis de consumidores da classe residencial e um terceiro agrupamento contendo os perfis de unidades consumidoras das classes comercial e industrial em baixa tensão.

8. Conclusões

A construção de tipologias de curvas de carga é uma etapa crítica nos processos de revisão das tarifas de uso dos sistemas de distribuição de energia elétrica (TUSD), pois envolve a realização de campanhas de medição visando a obtenção de registros das demandas horárias em unidades consumidoras e transformadores nas redes de baixa e média tensão. A partir dos registros obtidos em uma campanha de medição obtém-se uma base de dados contendo os perfis horários de demanda de todas as unidades consumidoras amostradas, cujas formas podem ser entendidas como manifestações ruidosas de alguns poucos perfis latentes que descrevem o comportamento típico da demanda por eletricidade ao longo do dia.

A identificação dos perfis latentes ou tipologias, fundamentais para o cálculo das tarifas de eletricide, pode ser realizada por meio de técnicas de análise de agrupamentos.

Neste trabalho são descritas três técnicas de *cluster analysis* (método de Ward, *K-Means* e K-Medoides) e apresentadas formas de implementá-las computacionalmente em ambiente R. Adicionalmente são discutidas as medidas de validação e estabilidade que permitem determinar o número adequado de agrupamentos.

Apesar de existirem *softwares* dedicados a construção de tipologias de curvas de carga ou programas comerciais para a análise de agrupamentos, a implementação em ambiente R tem

como vantagens o baixo custo, a variedade de métodos disponíveis para a análise de agrupamentos e a possibilidade de compará-los por meio de diversas medidas de validação e estabilidade.

Os programas apresentados no presente artigo constituem apenas um exemplo de como implementar a análise de agrupamentos no ambiente R. Contudo, destaca-se que os códigos descritos no artigo podem ser facilmente adaptados, por exemplo, com a inclusão de outros métodos para análise de agrupamentos não considerados neste trabalho em função das limitações de espaço.

Referências

BRASIL, Ministério das Minas e Energia, DNAEE, Eletrobrás, Empresas Concessionárias de Energia Elétrica (1985), Nova Tarifa de Energia Elétrica: metodologia e aplicação, Brasília: DNAEE.

Brock, G.; Pihur, V.; Datta, S. & Datta, S. (2008). clValid: An R package for Cluster Validation. Journal of Statistical Software, 25(4), 1-22.

Chantelou, D.; Hébrail, G. & Muller, C. (1996). Visualizing 2,665 electric power load curves on a single A4 sheet of paper. In: International Conference on Intelligent Systems Applications to Power Systems, Orlando.

Chicco, G.; Napoli, R. & Piglione, F. (2006). Comparisons Among Clustering Techniques for Electricity Customer Classification, IEEE Transactions On Power Systems, 21(2), 933-940.

Debrégeas, A. & Hébrail, G. (1998). Interactive interpretation of Kohonen maps applied to curves, In: International Conference on Knowlodge Discovery and Data Mining, New York.

Diday, E. (1971). Une nouvelle méthode em classification automatique et reconnaissance des formes. La méthode des nuées dynamiques. Revue de Statistique Appliquée, vol. XIV nº 2. Institut de Statistique. Université de Paris.

Everitt, B. (2007). An R and S –Plus Companion to Multivariate Analysis, London: Springer.

Frei, F. (2006). Introdução à análise de agrupamentos: teoria e prática, São Paulo: Editora UNESP.

Figueiredo, V.; Rodrigues, F.; Vale, Z. & Gouveia, J.B. (2005). An Electric Energy Consumer Characterization Framework Based on Data Mining Techniques, IEEE Transactions on Power Systems, 20(2), 596-602.

Gerbec, D.; Gasperic, S.; Smon, I. & Gubina, F. (2004). Determining the load profiles of consumers based on fuzzy logic and probability neural networks, Generation, Transmission and Distribution. IEE Proceedings, 151(3), 395-400.

Guardia, E.C.; Queiroz, A.R. & Marangon Lima, J.W. (2010). Estimation of Electricity Elasticity for Demand Rates and Load Curve in Brazil. In: IEEE Power and Energy Society General Meeting, Minneapolis.

Hartigan, J.A. & Wong, M.A. (1979). Algorithm AS 136: A k-means clustering algorithm. Applied Statistics, 28(1), 100-108.

Hébrail, G. (2001). Practical data mining in a large utility company. Questiió (Quaderns d'Estadistica i Investigació Operativa), 25(3), 509-520.

Jain, A.K.; Duin, R.P.W. & Mao, J. (2000). Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligent, 22(1), 4-37.

Jang, J.S.R.; Sun, C.T. & Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence. New Jersey: Prentice Hall.

Kaufman, L. & Rousseeuw, P.J. (1986). Clustering large datasets. In: Gelsema, E.S.; Kanal, L.N. (Eds.). Pattern Recognition in Practice II. Amsterdam: Elsevier/North-Holland 425–437.

Lattin,J.; Carrol, J.D. & Green, P.E. (2011). Análise de Dados Multivariados. São Paulo: Cengage Learning.

Leoni, R.C. & Costa, A.F.B. (2012). O ambiente R como proposta de apoio ao ensino no monitoramento de processos. Pesquisa Operacional para o Desenvolvimento, 4(1), 83-96.

Pessanha, J.F.M.; Velasquez, R.M.G.; Melo, A.C.G. & Caldas, R.P. (2002). Técnicas de cluster analysis na construção de tipologias de curva de carga. In: XV Seminário Nacional de Distribuição de Energia Elétrica, Salvador.

Pessanha, J.F.M.; Huang, J.L.C.; Pereira, L.A.C.; Passos Júnior, R. & Castellani, V.L.O. (2004). Metodologia e sistema computacional para cálculo das tarifas de uso dos sistemas de distribuição. In: XXXVI Simpósio Brasileiro de Pesquisa Operacional, São João del-Rei.

Pessanha, J.F.M.; Castellani, V.L.O. & Araújo, A.L.A. (2006). Uma nova feramenta computacional para a construção de tipologias de curva de carga. In: X Simpósio de Especialistas em Planejamento da Operação e Expansão Elétrica, Florianópolis.

Pessanha, J.F.M. & Laurencel, L.C. (2009). Clustering Electric Load Curves: The Brazilian Experience, In: Workshop Franco-Brésilien sur la Fouille des Données, Recife.

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Ramos, S.; Duarte, J.M.M.; Soares, J.; Vale, Z. & Duarte, F.J. (2012). Typical load profiles in the smart grid context – A clustering methods comparison. In: IEEE Power and Energy Society General Meeting, Porto.

Sathiracheewin, S. & Surapatana, V. (2011). Daily Typical Load Clustering of Residential Customers. In: 8th Electrical Engineering. Electronics, Computer, Telecommunications and Information Technology (ECTI), Bangkok.

Schrock, D.W. (1997). Load Shape Development. Oklahoma: PennWell Publishing Company.