

O PROBLEMA COMBINATÓRIO DE LOCALIZAÇÃO DE INSTALAÇÕES

Samuel Willian Alves Wu e Arnaldo Rabello de Aguiar Vallim Filho

Apoio: PIVIC Mackenzie

RESUMO

Este projeto contempla o problema de “localização de instalações”, que é um problema combinatório clássico de mineração de dados em que as instalações podem ser associadas a centroides de clusters. Do ponto de vista prático, uma de suas aplicações mais importantes está na área de Logística, onde representa a mais importante decisão estratégica. Neste projeto, o problema foi resolvido por rede neural artificial do tipo SOM – *self organizing maps*, combinada com um modelo matemático de otimização de programação binária. A rede SOM definiu clusters de pontos de demanda da operação e os centróides dos clusters foram definidos como locais candidatos a instalações. Pelo modelo matemático de otimização foi definido o número ótimo de instalações para atender a operação e os locais (centróides) em que essas instalações deveriam ser instaladas. O modelo matemático definiu também os pontos que deveriam ser alocados a cada instalação. O algoritmo combinado chegou a soluções satisfatórias para bases de dados de cidades brasileiras, mostrando-se com boa performance e atingindo os resultados esperados.

Palavras-chave: Redes neurais artificiais, localização, programação matemática

ABSTRACT

This project addresses the facility location problem, a classic combinatorial problem data mining, in which a facility may be a centroid of a cluster. From a practical point of view, one of its most important applications is in Logistics, in which it represents the most important strategic decision of an operation. In this project, the applied problem was solved by using artificial neural network techniques of the SOM type - self organizing maps, combined with a mathematical optimization model based on binary programming. The SOM network defined clusters of demand points and the centroid of each cluster was defined as a candidate for a facility location. The mathematical optimization model defined the optimal number and location of the facilities (the selected centroids) for the operation and the points that should be allocated to each facility. The combined algorithm found satisfactory solutions for Brazilian cities databases, showing a good performance and achieving the expected results.

Keyword: Artificial neural networks, location problem, mathematical programming

1. INTRODUÇÃO

Este trabalho é parte de um projeto mais amplo que busca o desenvolvimento de heurísticas e metaheurísticas para solução de problemas de otimização combinatória de diferentes tipos. No caso específico desta pesquisa, o foco é em problemas de localização de instalações (centroídes) para atendimento de pontos de demanda em uma operação logística.

Este é um conhecido problema logístico (Ballou, 2006) e é também um problema combinatório clássico da área de Mineração de Dados (Silva et al., 2016), em que as instalações são representadas por centroídes. Do ponto de vista prático, em Logística, representa a mais importante decisão estratégica de uma operação Logística (Ballou, 2006).

Neste projeto, pretende-se resolver este problema aplicado à área de Logística, por meio de técnicas de redes neurais artificiais (RNA) combinadas com um modelo matemático de otimização.

A importância do estudo de localizações é mostrada pelo grande número de aplicações que são encontradas em muitas áreas de atividade. O trabalho de Brandeau e Chiu (1989) ilustra essa aplicabilidade, com uma pesquisa em que mostra uma grande relação de aplicações, nas áreas privada e pública. Exemplos da área privada são: depósitos e centros de produção; terminais de transporte e centros logísticos de distribuição de carga; plantas industriais; redes de comunicações; centrais elétricas; centrais de táxi. Na área pública tem-se: centrais de veículos de emergência (ambulâncias, bombeiros, etc.); centros de serviços públicos em geral, tais como: centros de saúde, bancos de sangue, centrais de tratamento de lixo, centrais de tratamento de água; localização de bairros residenciais e até mesmo, instalações de defesa.

Crainic e Laporte (1997) mostram a importância estratégica da localização esclarecendo que as decisões estratégicas determinam, na verdade, o desenvolvimento geral das políticas de uma empresa e formatam de modo abrangente as estratégias operacionais do sistema sobre um relativamente longo horizonte de tempo e, dentre estas decisões estratégicas, está a localização de instalações.

Owen e Daskin (1998) afirmam que a localização de instalações é um aspecto estratégico crítico em uma grande variedade de atividades públicas e privadas e acrescentam ainda, que a decisão sobre uma nova instalação é tipicamente um projeto sensível ao tempo e ao custo. Localizações de instalações são definidas para permanecer em operação por um longo tempo. É assim, portanto, decisão estratégica

vital.

Ballou (2006) afirma que a localização de instalações pode ser considerada como o mais importante problema de planejamento logístico estratégico em um grande número de empresas. É esta decisão que estabelece as condições para a seleção apropriada e a boa administração dos serviços de transporte e dos níveis de estoque.

A importância da questão é, assim, bem reconhecida na literatura.

Sobre as abordagens de solução, os trabalhos mostram que na grande maioria dos casos a solução não se dá por técnicas de Inteligência Artificial, apesar de ser esta uma das áreas que mais cresce em computação (Vallim Filho, 2004).

Tratar assim o problema com uma abordagem diferenciada, em particular, com técnicas de uma área como a Inteligência Artificial, com grande destaque e crescimento no momento, pode trazer contribuições importantes para essa área do conhecimento.

1. 1 Problema de Pesquisa

Considerando-se, portanto, o problema da localização no contexto de uma operação logística, pode-se dizer que de forma básica, o problema contempla um conjunto de pontos distribuídos em uma área geográfica, que serão atendidos em sua demanda por instalações logísticas. Cada ponto tem uma demanda a ser satisfeita (medida em alguma unidade de peso, volume ou outra unidade) e cada instalação deverá atender um conjunto desses pontos.

Essas instalações logísticas podem ser:

- . Unidades de Produção (fábricas);
- . Centros de Armazenagem de Matéria Prima;
- . Centros de Armazenagem de Produto Acabado;
- . Centros de Distribuição;
- . Operadores Logísticos;
- . Outros

A literatura, por outro lado, apresenta diversos tipos de problemas de localização (*facility location problem*) e alguns autores apresentaram taxonomias para identificar os tipos e características desses problemas, como Klose e Drexl (2005) e ReVelle, Eiselt e Daskin (2008).

De forma geral, pode-se dizer que uma vez que a função da instalação esteja definida, a questão básica de localização de instalações envolve determinar ao menos:

Quantidade de instalações;

Localizações das instalações;

Alocação de pontos de demanda a cada instalação;

Nem sempre o foco do problema envolve todas essas decisões. Há problemas, por exemplo, que não consideram o abastecimento das instalações, como é o caso de postos de saúde ou postos de corpo de bombeiros. Já se as instalações forem, por exemplo, diferentes filiais de uma indústria, o abastecimento dessas Unidades de Produção, com matéria prima, a partir de fornecedores (fontes de fornecimento) é ponto chave a ser definido.

No caso específico deste projeto pretende-se trabalhar com um problema de localização que envolva as três decisões apresentadas acima.

Na maior parte dos casos, o problema da localização é resolvido tendo-se por objetivo a minimização de um índice que multiplica a distância percorrida pela quantidade de carga movimentada, ou então, pela minimização do custo logístico da operação.

Note-se que este é um problema de otimização combinatória, pois envolve uma grande quantidade de alternativas de solução (soluções viáveis), por isso, as técnicas de solução, em geral envolvem, metaheurísticas, já que este é um problema NP-Difícil, e assim, problemas de grande porte não podem ser resolvidos por algoritmos exatos, pois o tempo computacional cresce exponencialmente.

Nesta pesquisa pretende-se trabalhar com uma abordagem de solução baseada em RNA o que não é uma abordagem tão comum, já que a maioria das soluções propostas para esse tipo de problema se baseia em metaheurísticas. E na sequência, uma vez obtida a solução “ótima” do problema, propor uma apresentação gráfica para essa solução.

Assim, a questão que se coloca na pesquisa é:

“como se comporta a solução de um problema da localização quando resolvido por uma rede neural artificial (RNA) combinada com um modelo matemático de otimização, quais as dificuldades e facilidades de implementação dessas técnicas para solucionar esse tipo de problema, e como pode ser implementada uma apresentação gráfica da solução obtida?”

1. 2 Objetivos

O objetivo geral deste projeto é o desenvolvimento de um conjunto de experimentos para estudar o comportamento das soluções geradas para diferentes instâncias de um problema de localização.

A ideia básica da pesquisa é testar um tipo de solução com pouca utilização na literatura, que é a abordagem por RNA combinada com um modelo de otimização.

Em termos de objetivos específicos, pretende-se:

- . Desenvolver experimentos com diferentes instâncias utilizando-se bases de dados clássicas;
- . ajustar e treinar um algoritmo de otimização baseado em uma RNA do tipo SOM – *self organizing maps* ou mapas auto organizáveis, para solução do problema da localização;
- . combinar essa solução com um modelo matemático de otimização, baseado em programação binária;
- . apresentar a solução obtida por meio de visualizações com apoio de uma biblioteca gráfica.

2. REFERENCIAL TEÓRICO

Este capítulo tem como objetivo apresentar trabalhos científicos que possuem relação com o tema do presente projeto. Os materiais estudados auxiliaram no entendimento do problema e no conhecimento das estratégias de solução com o objetivo de tentar tratar a questão de localização de instalações.

O problema da localização tem sido resolvido por modelo exatos de Programação Matemática (PM) e por metaheurísticas (Vallim Filho, 2004). A PM é empregada para problemas de porte pequeno e médio, mas problemas de grande porte, que são os mais frequentes na prática são resolvidos por metaheurísticas, já que conforme dito acima, o problema é NP-Difícil.

O termo “metaheurística” deriva da composição de duas palavras gregas: “heurística” termo associado a um conhecimento circunstancial e o prefixo “meta” que significa “após”, indicando uma camada superior do processo de descoberta de uma solução para um problema. (Neto et al. 2010). Nesse sentido, metaheurísticas são técnicas que abrangem classes mais amplas de problemas. Assim, por exemplo: uma

mesma metaheurística que pode ser usada para resolver problemas de localização de instalações, pode também resolver problemas de roteirização de veículos. Uma metaheurística, na verdade, não é um método pronto para solucionar um problema. Estes métodos se constituem, na verdade, em “princípios gerais de busca organizados em estratégias gerais”, não podendo assim, ser considerados como algoritmos, mas sim realmente como métodos ou então como meta-algoritmos (Pirlot, 1996). Uma metaheurística pode ser considerada assim, como uma estratégia geral de solução de uma classe de problemas. Uma estrutura geral de solução, que uma vez implementada computacionalmente se torna um *framework*.

São algoritmos aproximados que incorporam mecanismos para evitar confinamento em mínimos ou máximos locais. Conhecimentos específicos do problema podem ser utilizados na forma de heurística para auxiliar no processo de busca (por exemplo, na busca de um possível bom vizinho de um determinado ponto). Em síntese, pode-se dizer que metaheurísticas são mecanismos de alto nível para explorar espaços de busca, cada uma usando um determinado tipo de estratégia (Becceneri, 2012).

Tratam-se de técnicas que estão na fronteira entre pesquisa operacional e inteligência artificial (Colomi et al., 1997).

Há uma quantidade grande de metaheurísticas. Laporte (2009) apresenta as principais categorias:

- . metaheurísticas de busca local, tais como: *simulated annealing*, *deterministic annealing* e *tabu search*
- . metaheurísticas de busca em população, tais como: *adaptive memory procedures*, *genetic search*
- . metaheurísticas de busca em população, tais como: *adaptive memory procedures*,
- . mecanismos de aprendizado, tais como: RNA e colônia de formigas.

Especificamente sobre o problema da localização, Vallim Filho (2004) mostrou que a localização ótima de Centros de Distribuição (CD) pode ser determinada através da metaheurística *Simulated Annealing* (SA). Para isto foi definida como função objetivo que minimiza os custos de localização e de transporte em uma rede com dois estágios transferência para CD's e entrega nos destinos finais.

Pirlot (1996) apresenta um tutorial para três metaheurísticas amplamente

utilizadas: SA, busca tabu e algoritmos genéticos. Uma descrição e aplicações das técnicas são apresentadas no artigo. Colorni et al. (1997) descreveram os principais aspectos das metaheurísticas derivadas da natureza, incluindo uma revisão e uma aplicação de SA. D'Amico et al. (2002) considerou o problema de redistribimento ou redesenho dos limites do comando policial através de metaheurística, e por conseguinte ficariam definidas as localizações das delegacias.

Bell (2003) desenvolveu uma pesquisa para construir um modelo para fornecer respostas sobre a melhor forma de reposição de inventários de munições da Força Aérea dos EUA necessários para futuros conflitos para uma variedade de cenários de demanda. Chiyoshi e Galvão (2000) desenvolveram uma análise estatística do SA aplicado ao conhecido problema da p-mediana, em que a mediana é a localização da instalação, a fim de identificar o desempenho da técnica. Eles testaram um conjunto de problemas, executando cada problema com no máximo cem fluxos diferentes de números aleatórios, o que é necessário para aplicar a metaheurística. Outro artigo interessante relacionado ao tema é o de Abdinnour-Helm (2001), que trata do problema p-hub, em que os hubs são as instalações, deve-se definir essas localizações para diferentes valores de p. O artigo mostra uma solução pela metaheurística SA.

Em outra publicação Silva e Cunha (2017) descrevem uma metaheurística de busca tabu (TS) para o problema de cobertura máxima do p-hub. O objetivo é determinar a melhor localização para p hubs e a atribuição de cada um dos pontos de demanda a um único hub, de modo que a demanda total entre pares de nós, dentro de uma determinada distância de cobertura, seja maximizada.

Em um dos poucos trabalhos baseados em RNA, Matsuda e Yoshimoto(2003) apresentam uma abordagem de solução por meio de RNA. O algoritmo parte de uma solução inicial e depois encontra a solução ótima por meio de uma rede baseada em mapas auto-organizáveis (*self organizing maps* – SOM).

Os mapas auto-organizáveis foram foco de Teuvo Kohonen em seu estudo na década de 1980 e são baseados sobretudo no princípio da formação de mapas de unidades cerebrais (neurônios), como apontado por Leandro Augusto (2016). De forma auto-organizada, estabelecem uma ordenação espacial que permite representar informação. A arquitetura é representada por duas camadas: uma composta por neurônios sensoriais sendo responsável pela entrada de dados e pelos estímulos enviados para a rede neural.

3. METODOLOGIA

3.1 Classificação da Pesquisa e Etapas da Metodologia

A pesquisa desenvolvida neste projeto é um tipo de pesquisa aplicada, já que trabalha com um problema eminentemente prático. A abordagem é quantitativa, e sua finalidade metodológica, pois o objetivo foi desenvolver um modelo computacional. Sobre os meios a utilizar no estudo, foi feita uma revisão bibliográfica do tema e da documentação das bibliotecas usadas.

O trabalho de pesquisa teve início por uma revisão bibliográfica e levantamento de bases de dados para os experimentos que seriam desenvolvidos. Em seguida, foram definidas de forma mais detalhada e final, as técnicas que seriam empregadas na solução do problema. Na sequência houve a necessidade de se definir ferramentas computacionais para implementação das técnicas. Finalmente, houve a fase de desenvolvimento e testes de algoritmos e de desenvolvimento de experimentos, onde foi possível a obtenção de resultados que puderam ser analisados.

Assim, pode-se dizer que a metodologia se desenvolveu segundo as etapas abaixo:

a) Revisão Bibliográfica

A revisão bibliográfica focou em trabalhos associados ao problema da localização, a técnicas de aprendizado de máquina, mais especificamente, redes neurais artificiais e a modelos de programação matemática

b) Levantamento de Bases de Dados

Os experimentos utilizaram bases de dados (*datasets*) disponíveis em sites científicos, procurando estudar em diferentes instâncias os resultados obtidos por uma solução obtida pela implementação de uma RNA do tipo SOM combinada com um modelo de programação binária.

c) Análise e detalhamento das técnicas a empregar no estudo

Definiu-se que o problema da localização seria resolvido por RNA do tipo SOM para clusterizar os pontos de demanda. E em seguida, seria aplicado um modelo de matemático de otimização para alocação dos pontos de demanda aos centróides dos clusters gerados pelo SOM

c) Seleção das ferramentas computacionais adequadas ao desenvolvimento

Trata-se da seleção das ferramentas computacionais empregadas no desenvolvimento dos algoritmos necessários para a solução do problema e da

apresentação de resultados. Para isto, foi definida a linguagem R especializada em técnicas estatísticas e de ciência de dados. Em R utilizou-se as bibliotecas RSNNS para a normalização dos dados dos datasets, a biblioteca kohonen para gerar a grade e o modelo SOM, a biblioteca pracma para realizar o cálculo de distâncias utilizando coordenadas e as bibliotecas ompr e magrittr para a construção de um modelo matemático de programação binária.

d) Seleção das bibliotecas gráficas para visualização de resultados

Todas as análises de resultados foram tratadas com técnicas de visualização e com procedimentos estatísticos formais por meio de bibliotecas gráficas de plotagem de dados. Foi usada a biblioteca gráfica do R, a ggplot2 e a biblioteca gráfica R-Shiny. Esta foi usada para fazer a apresentação gráfica, no formato de um Diagrama Estrela (centroídes conectados a pontos de atendimento).

e) Desenvolvimento dos algoritmos

Nesta etapa teve-se dois tipos de desenvolvimento:

- e₁) Construção de uma RNA do tipo SOM para clusterizar os pontos de demanda e definir candidatos a instalações nos centroídes dos clusters gerados
- e₂) Construção de um modelo matemático de otimização para definir localizações ótimas e alocar pontos de demanda às instalações

f) Experimentos e Análise de Resultados

Foram feitos testes com duas bases de dados diferentes, uma contendo as cidades dos Estados Unidos e a outra as cidades do Brasil, com sua latitude, longitude e população. Foi escolhida uma amostra dos *datasets* para que o estudo fosse direcionado a uma situação mais real a um problema de logística, sendo que para os Estados Unidos, são utilizados os dados das cidades do Estado de Delaware e para o Brasil, as cidades da região Sudeste e Nordeste. Nesta etapa, os algoritmos desenvolvidos puderam ser testados e gerar resultados, definindo localizações para instalações e a alocação de pontos de demanda a essas instalações.

3. 2 Descrição do Algoritmo

Em termos gerais, o algoritmo desenvolvido importou as bibliotecas usadas para a criação da grade e do modelo SOM, e para a normalização de dados. A função `som()` da biblioteca kohonen foi guardada em uma variável de nome `somFunc()`, pois a biblioteca RSNNS também possui uma função com mesmo nome, e ao importá-la, o R studio sobrescreve a função de kohonen.

A seguir, foi importada a base de dados que está no formato csv para o

ambiente do algoritmo, foram reunidas as cidades desejadas para o estudo e foram extraídas as variáveis de latitude, longitude e população para normalização dos dados, que foram armazenados em um data frame *data_train_matrix*.

Foi então criada para a RNA SOM uma grade do tipo hexagonal de dimensões 3x4, que é uma exigência do SOM para se dar início ao algoritmo. O modelo SOM foi criado a seguir, utilizando a função da biblioteca kohonen *som()*, que foi renomeada para *somFunc()*. Foi passado como parâmetro para a função: a base de dados já normalizada, a grade criada no passo anterior, o número de iterações realizadas, a taxa de aprendizado que foi deixada com o valor padrão, o argumento *keep.data*, que armazena as informações do modelo, e o raio, que também é uma exigência do SOM.

Depois da criação do modelo SOM, é possível verificar os valores que os neurônios adotam ao se organizar como centroides no modelo por meio do componente *codes*. Entretanto, os valores são gerados no padrão de normalização que foi usado no dataset, e por consequência, foi preciso passá-los para uma função também fornecida pela biblioteca RSNNS, que desempenha uma tarefa de normalização reversa. Dessa forma, é possível fazer o processo contrário de normalização, passando como parâmetro dados normalizados.

A seguir, foram criados datasets para armazenar os dados obtidos da análise. Os dados dos pontos de demanda, ou cidades foram armazenados em um *data frame* *customer_locations*, contendo um índice *id*, a latitude, longitude, população em sua forma original, sem passar pelo processo de normalização, e a localização, sendo esta o centroide à qual a cidade está atrelada. Já os centroides, ou armazéns, são armazenados no data frame *warehouse_locations*, possuindo os seguintes dados: um índice *id*; a latitude e longitude que passaram pelo processo reverso de normalização; variável *dist_to_mean*, que representa a distância entre cada centroide à média dos mesmos; *cost_per_square_meter*, o custo indicado para a localização do armazém, determinado por um vetor de distância entre os centroides e suas médias com o primeiro quartil representando o custo de 2000 reais por metro quadrado ($/m^2$), segundo quartil com $1500/m^2$, terceiro quartil com $1000/m^2$, e valores acima com $500/m^2$. O data frame também conta com a população somada de todas as cidades atreladas a cada centroide, o tamanho do armazém em metros quadrados determinado pelo número de habitantes, e o custo do armazém, sendo calculado multiplicando o custo por metro quadrado pelo seu tamanho.

O passo seguinte se dedica a criar uma matriz utilizada no modelo de programação matemática, contendo o custo de transporte de cada centroide para cada

cidade. O modelo de programação é criado nesse momento, utilizando como variáveis x_{ij} do tipo binário, representando se uma cidade “i” está atrelada a um centroide “j”, e a variável y_j também do tipo binário, representando se o centroide “j” será selecionado para receber uma instalação.

$$\text{Min } [\sum_i \sum_j \text{CustoTransporte}_{ij} \cdot x_{ij} + \sum_j \text{CustoFixo}_j \cdot y_j] \quad (3.1)$$

Sujeito a:

$$\sum_j x_{ij} = 1 ; i=1, \dots, n ; j=1, \dots, m \quad (3.2)$$

$$x_{ij} \leq y_j ; i=1, \dots, n ; j=1, \dots, m \quad (3.3)$$

$$x_{ij} \in \{0,1\} ; i=1, \dots, n ; j=1, \dots, m; y_j \in \{0,1\} ; j=1, \dots, m \quad (3.4)$$

O modelo é resolvido por meio do “solver” glpk da infraestrutura de otimização do R (R Optimization Infrastructure – ROI).

4. RESULTADO E DISCUSSÃO

O algoritmo obtido como resultado do processo descrito anteriormente foi capaz de gerar a solução gráfica esperada no começo do projeto. O algoritmo se resume as etapas apresentadas no capítulo de metodologia de pesquisa e serão mais detalhadas na sequência.

A figura 4.1 demonstra os passos referentes à importação: das bibliotecas para a criação do modelo de RNA do tipo SOM e a normalização de dados e da base de dados brasileira utilizada no algoritmo, filtrando apenas as cidades do Nordeste e do Sudeste. Foi necessário armazenar a função `som()` da biblioteca *kohonen* em uma variável de nome `somFunc()`, pois a biblioteca *RSNNS* também possui uma função de mesmo nome, sobrescrevendo a função que seria usada. A normalização dos dados é realizada a seguir, sendo armazenada na matriz `data_train_matrix`. Para a criação da grade do tipo hexagonal de dimensões 3x4, e do modelo, são utilizados os parâmetros *rlen* para definir o número de iterações, *alpha* para definir a taxa de aprendizado, *keep_data* para indicar que o modelo deve ser armazenado, e *radius* para definir o raio do modelo.

Figura 4.1 – Segmento de código de importação de bibliotecas

```
library(kohonen)
require(kohonen)
library(RSNNS)
somFunc <- kohonen::som

df <- read.csv('database/brcities.csv', header = TRUE, sep = ",")

#Base de dados com cidades proeminentes do nordeste, e sudeste
usedCities <- c(4:9, 30:60, 61:79, 81:85, 104:122, 141:183, 208:210, 231:242,
               243:249, 250:259, 260:266, 328:374, 375:376)
df_cities <- df[usedCities, c(2,3,8)]
rownames(df_cities) <- NULL

#data_train_matrix <- as.matrix(scale(df_cities))
data_train_matrix <- as.matrix(normalizeData(df_cities, type = "norm"))
colnames(data_train_matrix) <- c("lat", "lng", "population")

som_grid <- somgrid(xdim = 3, ydim = 4, topo="hexagonal") # SOM 3x4, hexagonal
som_model <- somFunc(data_train_matrix,
                    grid=som_grid,
                    nlen=300,
                    alpha=c(0.05,0.01),
                    keep.data = TRUE,
                    radius = 5)
```

A figura 4.2 mostra como é obtido o valor que os centroides adotam ao se organizar utilizando o atributo *codes* do modelo gerado, partindo para o processo de normalização reversa com a função *denormalizeData()*, fornecida pelo mesmo pacote usado para a normalização de dados RSNNS.

Figura 4.2 – Normalização reversa dos dados

```
#centroides de cada estado (12)
centroides <- as.data.frame(som_model$codes)
view(centroides)

#processo de denormalização.
centroides_norm <- as.data.frame(denormalizeData(centroides, getNormParameters(data_train_matrix)))
```

Depois disso, na figura 4.3 tem-se a importação da biblioteca gráfica *ggplot2*, onde são inicializadas as variáveis que indicam número de armazéns, número de cidades, a média da latitude e longitude dos armazéns.

Figura 4.3 – importação da biblioteca gráfica e inicialização de variáveis

```
library(ggplot2)
require(ggplot2)

m <- 12 #usado em warehouse locations
n <- 211 #usado em customer locations
D <- 0
x_mean <- mean(centroides[,1]) #media x dos centroides
y_mean <- mean(centroides[,2]) #media y dos centroides
```

Na figura 4.4 vê-se como é criado o *data frame customer_locations* contendo as cidades com sua latitude, longitude, A localização da cidade, sendo ela o centroide a qual a cidade está atrelada, e sua população. A localização foi retirada do atributo *unit.classif* do modelo som criado no início.

Figura 4.4 – Criação de *data frame customer_locations*

```
#data frame customer_locations
customer_locations <- data.frame(
  id = 1:n,
  x = df_cities[,1],
  y = df_cities[,2],
  localiz = as.matrix(som_model$unit.classif),
  population = df_cities$population$population
)
view(customer_locations)
```

Antes de se criar um *data frame* para os armazéns, é necessário realizar alguns cálculos para definir o custo de instalação. Para isso, é calculada a distância da média

para cada centroide(figura 4.5) e definidos os quartis, é possível distinguir as áreas de maior valor.

Figura 4.5 – Cálculo de distância e definição de quartis

```
#cálculo da distância entre cada centroide e a média dos centroides
for(val in 1:m){
  D <- distanc(centroides$lat[[val]], centroides$lng[[val]],
              x_mean, y_mean)

  centroiddistancevector[val] <- D
}
View(centroiddistancevector)

#def of quartiles of distances between centroids mean and its locations
quartile1 <- quantile(centroiddistancevector, 0.25)
quartile2 <- quantile(centroiddistancevector, 0.5)
quartile3 <- quantile(centroiddistancevector, 0.75)
```

O Custo de Localização é definido partindo-se da hipótese de que quanto mais próxima do centro urbano for uma área, maior será seu custo por metro quadrado. Assim, foram definidos quatro raios a partir do Centróide Médio, que corresponde á média dos centróides dos clusters gerados. Para cada raio estabelece-se um custo/m². Aquerles mais próximos do centróide média devem ter um custo/m² mais elvado e à medida que os raios crescem, o custo/m² diminui. Esses são parâmetros que podem ser modificados a cada experimento.

Para efeito do experimento que foi desenvolvido, foram definidos raios com base nos quartis das distâncias entre centróides de cada cluster e o centróide médio (Dc). Assim, foram utilizados os seguintes raios com centro no centróide médio:

- . Raio 1: $D_c \leq Q$ → Custo de 2.000,00/m²
 - . Raio 2: $Q_1 < D_c \leq M_d$ → Custo de 1.500,00/m²
 - . Raio 3: $M_d < D_c \leq Q_3$ → Custo de 1.000,00/m²
 - . Raio 4: $M_d \geq Q_3$ → Custo de 500,00/m²
- onde: Q_1 = 1º quartil de Dc
 Q_3 = 3º quartil de Dc
 M_D = Mediana de Dc

Note-se que estes valores de custos representam um dado cenário para se proceder a um experimento. Vários cenários podem ser montados para se analisar o impacto desses custos na solução. A figura 4.6 mostra o segmento de código correspondente a esse procedimento baseado em quartis.

Figura 4.6 – Definição de custo por metro quadrado

```
for(val in 1:m){
  if(centroiddistancevector[val] <= quartile1){
    centroid_costPerSquareMeter[val] <- 2000 #custo por metro quadrado
  }
  if(centroiddistancevector[val] > quartile1 && centroiddistancevector[val] <= quartile2){
    centroid_costPerSquareMeter[val] <- 1500
  }
  if(centroiddistancevector[val] > quartile2 && centroiddistancevector[val] <= quartile3){
    centroid_costPerSquareMeter[val] <- 1000
  }
  if(centroiddistancevector[val] > quartile3 ){
    centroid_costPerSquareMeter[val] <- 500
  }
}
View(centroid_costPerSquareMeter)
```

Com uma estrutura de repetição, cada armazém é separado dependendo de sua localização, sendo possível calcular o custo por metro quadrado. Para armazéns que estiverem mais distantes da área de maior concentração, o preço por metro quadrado será menor, com 500 reais por metro quadrado. Para os que estiverem mais ao centro, o custo será de 2000 reais por metro quadrado. Definidos os custos unitários das áreas de armazém, a dimensão de cada armazém foi estabelecida de forma proporcional à população de cada região. A área de cada armazém é calculada multiplicando-se a soma das populações das cidades alocadas ao armazém, multiplicada por um parâmetro de m^2 de armazém por habitante, conforme o seg.emtop de código apresentado na figura 4.7.

Figura 4.7- Definição de população total, custo e área dos armazéns

```
#soma a população de cada centroides
clustPop <- vector(length = m)
for(i in 1:m){
  for(j in 1:n){
    if(customer_locations$localiz[j] == i){
      clustPop[i] <- clustPop[i] + customer_locations$population[j]
    }
  }
}
view(clustPop)

#calc of warehouse size and cost
warehouse_costs <- vector(length = m)
warehouse_size <- vector(length = m)
meter_per_habitant <- 1
for(i in 1:m){
  warehouse_size[i] <- clustPop[i] * meter_per_habitant
  warehouse_costs[i] <- warehouse_size[i] * centroid_costPerSquareMeter[i]
}
```

A *data frame* de armazéns é criada com informações de latitude, longitude, distância dos armazéns para o centróide médio, custo por metro quadrado, população total, área em metros quadrados e custo fixo (figura 4.8).

Figura 4.8 – Criação de *data frame warehouse_locations*

```
warehouse_locations <- data.frame(
  id = 1:centroid_id,
  x = centroides_norm$V1,
  y = centroides_norm$V2,
  dist_to_mean = centroidDistancevector, #dist of each warehouse to all warehouse mean
  cost_per_square_meter = centroid_costPerSquareMeter, #cost based on dist_to_mean
  total_population = clustPop,
  warehouse_size = warehouse_size, #size based on population
  warehouse_costs = warehouse_costs #cost based on warehouse_size and cost_per_square_meter
)
view(warehouse_locations)
```

No que diz respeito à base de custos de transporte necessários para alimentar o modelo, utilizou-se a metodologia de cálculo de fretes, proposta por Vallim Filho e Kurihara (2001) e pela Agência Nacional de Transportes Terrestres (ANTT), na resolução número 5.820, que instituiu a Política de Preços Mínimos do Transporte Rodoviário de Cargas, e que foi emitida em decorrência da recente greve dos caminhoneiros ocorrida no país, no mês de maio de 2018. A seguir, é possível visualizar os procedimentos de cálculo de custos fixos (tabela 2.1) e custos variáveis (tabela 2.2) de um veículo. Estes custos definiram os parâmetros a serem utilizados no experimento. Os cálculos foram baseados na metodologia da ANTT (2018).

Tabela 2.1 – Planilha de Custos Fixos do Veículo

CUSTOS OPERACIONAIS DE VEÍCULOS			
PLANILHA ANTT			
Custos Fixos			
1. Reposição do Veículo		5. Tributos sobre o Veículo	
Vr. do Veículo	230.000,00	Vr. do Veículo	230.000,00
Valor Residual do Veículo	60%	% de IPVA	3%
Perda durante a Vida Útil	92.000,00	Custo Anual de IPVA	6.900,00
Vida Útil (meses)	72	DPVAT + Licenciamento	200,00
Custo Mensal de Reposição do Veículo	1.277,78	Custo Anual (c/ DPVAT e Licenciamento)	7.100,00
2. Reposição do Equipamento de Carga		Custo Mensal de Tributos	591,67
Vr. do Equipamento de Carga	35.000,00	6. Riscos de Acidentes e Roubo do Veículo	
Valor Residual do Equipamento de Carga	10%	Importância Segurada	230.000,00
Perda durante a Vida Útil	31.500,00	% Seguro do Veículo	3%
Vida Útil (meses)	84	Custo Anual	6.900,00
Custo Mensal de Reposição do Equipam.	375,00	Custo Mensal de Riscos do Veículo	575,00
3. Remuneração Mensal de Capital		7. Riscos de Acidentes e Roubo do Equipamento de Carga	
Valor do Veículo	220.000,00	Importância Segurada	35.000,00
Valor do Equipamento de Carga	35.000,00	% Seguro do Equipamento de Carga	6,5%
Vr. do 3o Eixo	10.000,00	Custo Anual	2.275,00
Total	265.000,00	Custo Mensal de Riscos do Equipamento	189,58
Taxa Anual de Juros	12%	Custo Fixo Mensal Total	
Custo Mensal de Capital	2.650,00	9.934,03	
4. Custos de Mão de Obra de Motorista			
Salário Nominal de Motorista	1.500,00		
Reserva de Encargos Sociais	90%		
No. de Motoristas por Veículo	1,5		
Custo Mensal de Motorista	4.275,00		

Os custos fixos do veículo são calculados em uma base mensal e são expressos em R\$/mês. Os custos variáveis do veículo são calculados em uma base quilométrica, sendo expressos em R\$/km. Todos os parâmetros necessários para o cômputo desses custos foram levantados no mercado, tais como: preços de veículos, preços de combustível, salários, etc. Para cômputo desses custos foi considerado um veículo médio (capacidade de 12t) cujas características e custos de insumos foram obtidos em fornecedores e portais do mercado de transportes de carga. A tabela 2.3 apresenta o resumo final desses custos.

Para o cálculo do custo de transporte é utilizado o custo final calculado na planilha da tabela 2.3 e é necessária também a função haversine da biblioteca pracma que recebe as coordenadas geográficas medidas em graus, e retorna a distância entre as duas em quilômetros. O valor é multiplicado pelo resultado do cálculo por km (tabela 2.3), que foi arredondado para R\$2,50, e pela volume do armazém em toneladas. A função que chamamos de transportCostMatrixFact realiza todos os cálculos e gera a matriz de custo de transporte. O custo de transporte total da operação considera toda a tonelagem que será movimentada em cada armazém e que deverá ser transportada.

Isto é feito pela equação 4.1:

$$CTr = Dc \cdot (C_{KM}/Cap) \cdot (A.h/p) \quad (4.1)$$

onde:

CTr = custo de transporte total em uma rota

Dc = distância do centróide (cidade) até o centróide médio

C_{KM} = custo por km do veículo padrão de transporte

Cap = capacidade do veículo

A = área do armazém

h = pé direito do armazém

ρ = densidade média da carga

Tabela 2.2 – Planilha de Custos Variáveis do Veículo

CUSTOS OPERACIONAIS DE VEÍCULOS			
PLANILHA ANTT			
Custos Variáveis			
1. Manutenção			
Vr. do Veículo + Equipamentos	265.000,00		
Índice de Manutenção	1,0%		
Custo Mensal de Manutenção	2.650,00		
Km Média Mensal do Veículo	8.000		
Custo de Manutenção / Km	0,33		
2. Combustível			
Preço do Combustível (R\$/l)	3,00		
Rendimento (Km/l)	5,0		
Custo de Combustível / Km	0,60		
3. Lubrificantes			
Óleo de Motor			
Preço por Litro de Óleo	30,00		
Capacidade do Câter	14		
Litros por Reposição	1		
No. de Reposições	9		
Total de Litros (Troca + Reposições)	23		
Custo Total (Troca + Reposições)	690,00		
Km de Troca	10.000		
<i>Custo de Óleo de Motor / Km</i>	<i>0,069</i>		
Óleo de Transmissao			
Preço por Litro de Óleo	40,00		
Capacidade do Câmbio + Diferencial	20		
Custo Total	800,00		
Km de Troca	20.000		
<i>Custo de Óleo de Transmissao / Km</i>	<i>0,0400</i>		
Custo de Óleos / Km	0,109		
4. Lavagem e Graxas			
Preço da Lavagem			300,00
Km entre Lavagens			4.000
Custo de Lavagens / Km			0,08
5. Pneus e Recauchutagens			
Preço de um Pneu			1.000,00
Preço de Câmara			75,00
Preço de Protetor			25,00
Preço Total de um Pneu			1.100,00
Perda de Pneus (Acidentes)			3%
Custo de Pneu com Perda			1.133,00
Preço de Reforma de Pneu			250,00
No. de Reformas por Pneu			2
Custo de Reformas de Pneu			500,00
No. de Câmaras Adicionais por Pneu			1
Custo de Câmaras Adicionais por Pneu			75,00
No. de Protetores Adicionais por Pneu			1
Custo de Protetores Adicionais por Pneu			25,00
Custo Total por Pneu			1.733,00
No de Pneus por Veículo			11
Custo Total de Pneus por Veículo			19.063,00
Vida Útil de Pneus			150.000
Custo de Pneus / Km			0,127
Custo Variável Total / Km			1,24

Tabela 2.3 – Planilha Resumo dos Custos Fixos e Variáveis do Veículo

Km por Mês	8.000
C. Variável por Km	1,24
C. Variável Mensal	9.938,69
C Fixo Mensal	9.934,03
C. Mensal TOTAL : Fixo + Variável	19.872,72
C. KM TOTAL : Fixo + Variável	2,48

A distância entre cada cidade e o armazém é multiplicada pela divisão do custo por km do veículo dividido pela capacidade do veículo transporte (resulta em R\$/ton.km). O resultado é multiplicado pela tonelagem total da região, calculada e pela área do armazém em metros quadrados, multiplicada pelo pédirito (altura) do armazém e dividida pela densidade média da carga brasileira (300kg/m³ – Fonte: ANTT)). A figura 4.9 apresenta o segmento de código em que esses cálculos são desenvolvidos.

Com a matriz de custo de transporte criada, é possível modelar o problema, utilizando aplicando-se um modelo de otimização, pela função `MIPModel()` da biblioteca `ompr` (figura 4.10). São adicionadas as variáveis `x` e `y`, representando de forma binária se uma cidade “`i`” está atrelada a um armazém, e se um armazém “`j`” será construído ou não.

Figura 4.9 – Cálculo do custo de transporte

```
#haversine
library(pracma)
require(pracma)

#transport cost calculation
transportcost_func <- function(i, j) {
  customer <- customer_locations[i, ]
  warehouse <- warehouse_locations[j, ]
  # calcula o custo de transporte
  return(haversine(c(customer$x, customer$y), c(warehouse$x, warehouse$y))
    * (2.5/25) * (warehouse$warehouse_size * 12/0.3))
}
transportcost_func(1,7)

transportCostMatrixFact <- function(){
  transport_cost <- matrix(nrow = n, ncol = m)

  for(row in 1:n){
    for(col in 1:m){
      transport_cost[row, col] <- transportcost_func(row, col)
    }
  }

  return(transport_cost)
}
transport_cost <- as.data.frame(transportCostMatrixFact())
```

A função de maximização é definida para priorizar opções baseando-se na soma do custo de transporte e no custo de instalação de armazéns. As restrições indicam que todas as cidades devem ser atreladas a um armazém, e se um armazém tiver pelo menos uma cidade para atender, ele deverá ser construído.

Figura 4.10 – Criação do modelo matemático

```
#solving model
library(ompr)
library(magrittr)
#masked functions: and, mod, or

model_MIP <- MIPModel() %>%
  # 1 iff i gets assigned to warehouse j
  add_variable(x[i, j], i = 1:n, j = 1:m, type = "binary") %>%

  # 1 iff warehouse j is built
  add_variable(y[j], j = 1:m, type = "binary") %>%

  # maximize the preferences
  set_objective(sum_expr(transportcost_func(i, j) * x[i, j], i = 1:n, j = 1:m) +
    sum_expr(warehouse_costs[j] * y[j], j = 1:m), "min") %>%

  # every customer needs to be assigned to a warehouse
  add_constraint(sum_expr(x[i, j], j = 1:m) == 1, i = 1:n) %>%

  # if a customer is assigned to a warehouse, then this warehouse must be built
  add_constraint(x[i, j] <= y[j], i = 1:n, j = 1:m)
model_MIP
```

Resolvido o modelo utilizando o solucionador `glpk` da infraestrutura de otimização do R (R Optimization Infrastructure – ROI), os resultados são armazenados na variável `matching` (figura 4.11)

Figura 4.11 – Resolução do modelo matemático

```
library(ompr.roi)
library(ROI.plugin.glpk)
result <- solve_model(model_MIP, with_ROI(solver = "glpk", verbose = TRUE))

suppressPackageStartupMessages(library(dplyr))
matching <- result %>%
  get_solution(x[i, j]) %>%
  filter(value > .9) %>%
  select(i, j)
```

Os *plots* são realizados utilizando dados dos data frames *warehouse_locations* e *customer_locations* e o resultado do modelo matemático (figuras 4.12a e 4.12b). No gráfico, é possível ver a distribuição dos armazéns seguido do custo fixo de montagem, e da quantidade de cidades que ele terá que atender.

Figura 4.12b– Plotagem dos dados apresentados – segmento I de código

```
grid_size <- 0
#principal PLOT
p <- ggplot(customer_locations, aes(x, y)) +
  geom_point() +
  geom_point(data = warehouse_locations, color = "red", alpha = 0.5, shape = 17) +
  scale_x_continuous(limits = c(-25, 25)) +
  scale_y_continuous(limits = c(-53, 33)) +
  theme(axis.title = element_blank(),
        axis.ticks = element_blank(),
        axis.text = element_blank(), panel.grid = element_blank())
p + ggtitle("warehouse location problem",
           "Black dots are customers. Light red triangles show potential warehouse locations.")
```

Figura 4.12b– Plotagem dos dados apresentados – segmento II de código

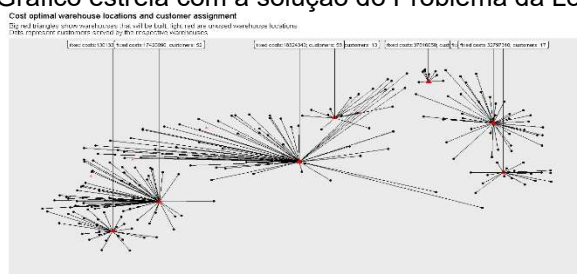
```
#add the assignments to the previous plot
plot_assignment <- matching %>%
  inner_join(customer_locations, by = c("i" = "id")) %>%
  inner_join(warehouse_locations, by = c("j" = "id"))
customer_count <- matching %>% group_by(j) %>% summarise(n = n()) %>% rename(id = j)

##### problema com fixed cost (o custo fixo deste código varia)
#armazéns escolhidos
plot_warehouses <- warehouse_locations %>%
  mutate(costs = warehouse_costs) %>%
  inner_join(customer_count, by = "id") %>%
  filter(id %in% unique(matching$j))

p +
  geom_segment(data = plot_assignment, aes(x = x.y, y = y.y, xend = x.x, yend = y.x)) +
  geom_point(data = plot_warehouses, color = "red", size = 3, shape = 17) +
  ggrepel::geom_label_repel(data = plot_warehouses,
                           aes(label = paste0("Fixed costs:", costs, "; customers: ", n)),
                           size = 3, nudge_y = 20) +
  ggtitle(paste0("Cost optimal warehouse locations and customer assignment"),
         "Big red triangles show warehouses that will be built, light red are unused warehouse locations. Dots represent customers served by the respective warehouses.")
```

A figura 4.14 apresenta o resultado final gerado pelo algoritmo apresentado na forma de um gráfico estrela com a solução do Problema da Localização. Cada armazém é conectado aos seus pontos de atendimento.

Figura 4.14 – Gráfico estrela com a solução do Problema da Localização



5. CONSIDERAÇÕES FINAIS

O projeto teve sucesso em buscar o retorno gráfico e o cálculo dos dados referentes aos armazéns gerados artificialmente pela rede neural SOM. Como apontado pelo trabalho de Brandeau e Chiu (1989), a pesquisa mostra uma grande relação de aplicações, nas áreas privada e pública, e pode ser reaproveitada para um conjunto de novos trabalhos e pesquisas, visto que o projeto opta por uma forma entre várias de se buscar a melhor localização de instalações. Etapas posteriores podem

ser objeto de estudo, como a etapa 1 do projeto que contemplava a busca pela melhor definição de rotas, tendo as instalações já definidas.

É possível desenvolver novos estudos para otimizar o algoritmo, buscando o uso de maiores quantidades de dados, por exemplo. No caso deste projeto, são utilizadas apenas as variáveis de latitude, longitude e população para a definição de rotas, abrindo opções para usarmos mais variáveis na definição de instalações, como a densidade demográfica das cidades, ou até a frequência de demanda de uma certa região. A aplicação do algoritmo em bases de dados diferentes também é uma opção, abrindo a oportunidade de verificar a eficácia do algoritmo com dados diferentes.

REFERÊNCIAS

- ANTT (2019). RESOLUÇÃO nº 5.849, de 16 de julho de 2019. Estabelece as regras gerais, a metodologia e os coeficientes dos pisos mínimos, referentes ao quilômetro rodado na realização do serviço de transporte rodoviário remunerado de cargas, por eixo carregado, instituído pela Política Nacional de Pisos Mínimos do Transporte Rodoviário de Cargas - PNPM-TRC.
- BALLOU, R.H. (2006) Gerenciamento da cadeia de suprimentos/ Logística empresarial, 5 ed., Bookman Editora. Porto Alegre, RS.
- BECCENERI, J.C., (2012) Metaheurísticas e Otimização Combinatória: Aplicações em Problemas Ambientais. 2 ed..
- BRANDEAU, M. L., CHIU, S.S. (1989). "An overview of representative problems in location research"; *Management Science*, vol. 35 , n.6, p.645-674.
- COLORNI,A., DORIGO, M., MAFFIOLI, F., MANIEZZO, V., RIGHINI, G., TRUBIAN, M. (1997). "Heuristics from nature for hard combinatorial optimization problems"; *International Transactions in Operational Research*, No.3.1 ,p. 1 - 38.
- CRAINIC, T.G. e LAPORTE, G. (1997) "Planning models for freight transportation. " *European Journal of Operational Research* 97, p. 409-438.
- DULLAERT, W.; VERNIMMEN, B.; WITLOX, F. (2002) A hybrid approach to designing inbound-resupply strategies. v.20.
- GITMAN, L. J. (2010). Princípios de Administração Financeira. 12ª Edição. Pearson Universidades. São Paulo, SP. 800p.
- HAYKIN, S. Neural Networks and Learning Machines. 3ª ed, Editora Prentice-Hall, 2008.
- KLOSE,A.; DREXL, A. "Facility location models for distribution system design. (2015) " *European Journal of Operational Research*.162 - 4–29.
- LUKE, S. Essentials of Metaheuristics. 2 ed, p. 9-12, 2015.

- MARTINS, E. (2018). Contabilidade de Custos - 11ª Edição. Editora Atlas. São Paulo, SP. 408p.
- MATSUDA, A.; YOSHIMOTO, K. (2003). On the use of neural networks to solve location problems: Application to the long-term and chronological model. DSI & APDSI, Shanghai
- MICHALEWICZ, M.; SCHMIDT, M.; MICHALEWICZ, Z.; CHIRIAC, C. (2005) Case study: an intelligent decision support system. v.20.
- NETO, L. B.; BECCENERI, J. C.; SILVA, J. D. S.; LUZ, E. F. P.; NETO, A. J. S. (2010) Fundamentos de Otimização e Inteligência Artificial. n. 4, p. 35-37.
- NTC & Logística. (2014) Manual de cálculo de custos e formação de preços do transporte rodoviário de cargas – 2014. Associação Nacional do Transporte de Cargas e Logística. São Paulo, SP.
- OWEN, S.H. e DASKIN, M. S. (1998) "Strategic facility location: a review. " European Journal of Operational research. No. 111, 423-447.
- PIRLOT, M. (1996). "General local search methods ". European Journal of Operational Research. No. 92, p.493-511.
- ReVELLE, C.S.; EISELT, H.A.; DASKIN, M.S. (2008) A bibliography for some fundamental problem categories in discrete location science. European Journal of Operational Research 184: 817–848.
- SILVA, L. A.; PERES, S. M.; BOSCARIOLI, C. (2016). "Introdução à Mineração de Dados: Com Aplicações em R". 1ª Ed. Elsevier Editora Ltda. R.J. 277 p.
- SILVA, M. R. ; CUNHA, C. B. (2017). A tabu search heuristic for the uncapacitated single allocation p-hub maximal covering problem. European Journal of Operational Research 262, (3), 954-965
- VALLIM FILHO, A. R. A., KURIHARA, T. (2001). Um modelo matemático de decisões estratégicas para a logística de distribuição de produtos. Revista Mackenzie de Engenharia e Computação: v.2, p.42 - 62. São Paulo, SP.
- VALLIM Fo , A. R. A. (2004) Localização de centros de distribuição de carga: contribuições à modelagem matemática. Tese de Doutorado, Escola Politécnica da Universidade de São Paulo. São Paulo, 286p..
- VENABLES, W. N.; SMITH, D. M. and the R Core Team. (2016). An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics. Version 3.3.2 (2016-10-31)

Contatos: samuelwuw@hotmail.com e aavallim@mackenzie.br