**Modelagem de Problema de Roteamento de Veículos com o pacote RouteR**

# *Luciane Ferreira Alcoforado-Academia da Força Aérea*

# Resumo.

Este trabalho apresenta o pacote RouteR como auxiliar no processo de modelagem de Problemas de Roteamento de Veículos até a obtenção da solução ótima utilizando o algoritmo de Programação Inteira.

**Palavras-chave:** VPR, Vehicle Routing Problem, RouteR

# Abstract

.

**Keywords:** . VPR, Vehicle Routing Problem, RouteR

# Introdução

O problema de roteamento de veículos (VRP - Vehicle Routing Problem) é um desafio de otimização logística que envolve determinar as rotas mais eficientes para um conjunto de veículos que devem atender a uma série de locais de destino, chamados de clientes, levando em consideração diversas restrições e objetivos. Tradicionalmente, o objetivo é minimizar os custos totais, como o tempo de viagem, a distância percorrida, o custo de combustível, ou maximizar a eficiência da operação.

As principais características do VRP incluem:

1. **Depósitos e clientes:** Existem um ou mais depósitos de onde os veículos partem e retornam após a conclusão das rotas, além de clientes que devem ser atendidos.
2. **Capacidade dos veículos:** Cada veículo possui uma capacidade limitada para transportar mercadorias, o que impõe uma restrição à quantidade de itens que podem ser entregues em cada rota.
3. **Distâncias e tempos de viagem:** As distâncias entre os locais e os tempos de viagem podem variar, impactando diretamente nos custos e na eficiência das rotas.
4. **Restrições operacionais:** Podem incluir restrições de tempo, como janelas de tempo em que os clientes podem ser atendidos, restrições de precedência (alguns clientes devem ser visitados antes de outros), entre outras.

De acordo com Toth & Vigo (2002), existem várias variantes (famílias) do VRP, cada uma com suas próprias características e complexidades, como o CVRP (*Capacitated Vehicle Routing Problem*) com restrições de capacidade de carga, o VRP com janelas de tempo (VRPTW – *VRP with time windows*), o VRP com capacidade de carga e tempo (CVRPTW ), o VRP com frota heterogênea (HFVRP - *Heterogeneous Fleet VRP*), entre outros. O objetivo principal em todos os casos é encontrar um conjunto de rotas ótimas que atendam a todas as restrições e objetivos definidos.

O sucesso da utilização de técnicas de otimização se deve não apenas à potência dos atuais sistemas informáticos e à plena integração dos sistemas de informação nas operações e processos comerciais, mas também pode ser atribuído ao desenvolvimento de modelos matemáticos rigorosos, capazes de levar em conta quase todas as características do VRP decorrentes de aplicações do mundo real. Além disso, os algoritmos correspondentes e suas implementações de computador (ferramentas de software) desempenham um papel essencial na busca de soluções viáveis de alta qualidade para instâncias do mundo real dentro tempos de computação aceitáveis. Em comparação com procedimentos não baseados em técnicas de otimização, podem ser alcançadas economias significativas de custos e uma melhor utilização da frota de veículos. (Toth & Vigo, 2002).

A experiência desta autora mostra que não basta ter conhecimento de um vasto conjunto de algoritmos que solucionam os problemas, mas sim é preciso dispor de meios práticos para que o usuário comum possa se apropriar de tais ferramentas e obter a solução desejada, promovendo a saúde financeira de instituições tanto públicas como privadas.

Particularmente no caso em questão quanto ao roteamento de veículos, a primeira dificuldade do usuário é informar ao software os dados necessários para o emprego do(s) algoritmo(s) adequado(s). Diante desta situação, surge a pergunta que este artigo pretende responder: **“Como auxiliar o usuário a desenvolver de maneira prática os dados necessário para a aplicação do algoritmo de Programação Inteira (IPA) no ambiente computacional R?”**

# Objetivos

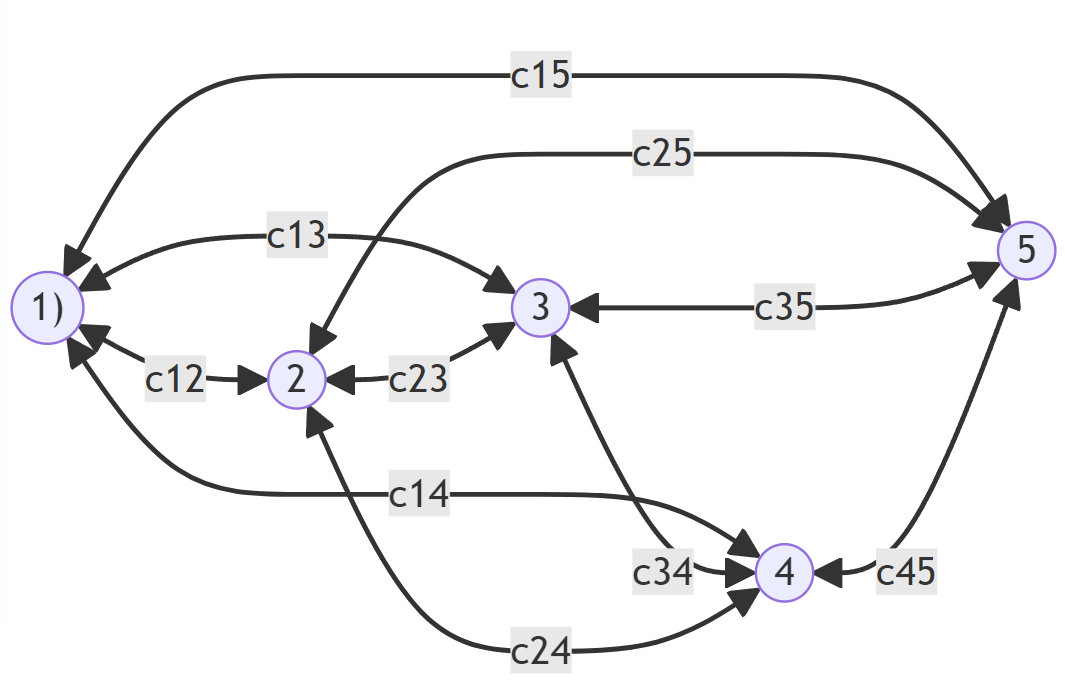
Objetivo Geral: Apresentar o Pacote RouteR para auxiliar a modelagem de Problema de Roteamento de Veículos (VRP).

Objetivos Específicos: construir a função objetivo; gerar as restrições estruturais do VRP; obter a solução ótima do VRP.

# Material e Método

# A notação básica adotada neste trabalho é descrita a seguir:

Seja G = (V, A) um grafo direcionado em que V representa o conjunto de vértices (clientes) e A representa o conjunto de arcos (trecho de ligação entre dois vértices), sendo V = {1, 2, ..., n} e A = (V x V), veja um exemplo na Figura 1. Considere K o conjunto de veículos disponíveis, K = { 1, 2, ..., |K|}

**Figura 1 –** Grafo completo com V = {1,2,3,4,5} e A = {(1,2), (1,3), …, (5,4)}, cij representa o custo do arco (i,j) em qualquer sentido.

Fonte: Autora, 2024.

A variável de decisão é definida por xijk = 1 se e somente se o veículo k se move pelo arco (i,j) ϵ A, sendo k o índice que representa o veículo, k = 1, 2, ..., |K|.

O modelo de otimização linear com o objetivo de minimizar o custo total da rota é dado por:

Min Z = , (Eq. 1)

*Sujeito a:*

1. Família de restrições que garantem que cada veículo saia do ponto inicial 1.

, , k = 1, 2, ..., |K| (Eq. 2)

1. Família de restrições que garantem que cada veículo retorne ao ponto inicial 1.

, , k = 1, 2, ..., |K| (Eq. 3)

1. Família de restrições que garantem que cada cliente seja visitado uma única vez por algum dos veículos.

, , i = 1, 2, ..., n (Eq. 4)

1. Família de restrições que garantem que nenhum cliente retenha o veículo que o visita.

, , i = 1, 2, ..., n e k = 1, 2, ..., |K| (Eq. 5)

1. Família de restrições que garantem a eliminação de sub-rotas de comprimento |S|-1, em que |S| representa o número de vértices na sub-rota, contido em A.

, , k = 1, 2, ..., |K| (Eq. 6)

1. Define a variável de decisão como binária

(Eq. 7)

A modelagem apresentada nas equações Eq.1 a Eq.7 geram uma quantidade grande de restrições, que cresce exponencialmente a medida que as restrições de impedimento de sub-rotas são adicionadas.

Considerando um grafo completo em que todos os vértices se conectam entre si, o número de variáveis de decisão do problema será da ordem de (n2 – n)k, assim, para um problema com n=5 e k=2 haverá precisamente (25 – 5)\*2 = 40 variáveis binárias, com n=9 e k=5 serão 360 variáveis binárias. Cada família de restrição produzirá a seguinte quantidade de restrições: I. e II. |K|, assim com k=5 serão 5 restrições em cada categoria; III. n, assim com n=9 vértices serão 9 restrições; IV. nk, assim com n=9 e k=5 serão 45 restrições; V. , assim, com n=9 e k=5 e |S|=2, serão 5\*(5!/(2!3!)) = 50 restrições, portanto um problema com n=9, k=5 e |S|=2 haverá 360 variáveis e 124 restrições, tal modelo garante evitar sub-rotas de comprimento 1. A cada nova família de restrição de sub-rota o problema vai aumentando o número de restrições podem alcançar o máximo de xx restrições, evitando-se sub-rotas de comprimento 1, 2, 3, ..., 7.

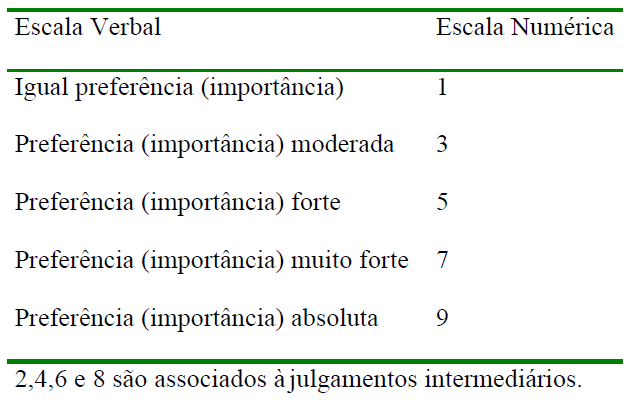
A metodologia adotada neste trabalho considerou a observação do modelo apresentado, produzindo as funções para gerar cada família de restrições utilizando-se da linguagem computacional R e a lógica de programação.

# O pacote RouteR

O pacote foi criado a partir da dificuldade enfrentada pela autora na elaboração do modelo de programação linear uma vez que gerar todas as restrições pode ser um grande desafio enfrentado por muitos pesquisadores ao aplicar a teoria a um problema real.

A seguir as funções desenvolvidas para o pacote, para um problema com n nós, sendo o nó 1 considerado o nó de origem e destino final, enquanto que os nós 2, …, n – 1 são os nós a serem visitados uma única vez. Todas as funções desenvolvidas no pacote RouteR (versão xx) baseiam-se nas equações apresentadas anteriormente Eq. 1 a Eq. 7.

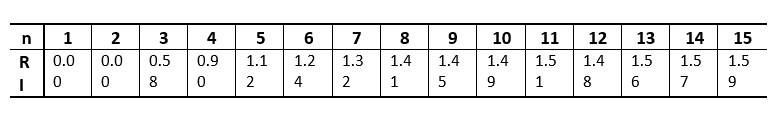
**Tabela 1 –** Escala de julgamento de Saaty



**Fonte:** Costa, 2002

A Tabela 2 mostra a tabela RI contendo os índices randômicos calculados pelo laboratório Oak Ridge para matrizes recíprocas quadradas de ordem n.

**Tabela 2 –** Índices de consistência aleatória



**Fonte:** Saaty, 1991

# Criação do pacote

Foram utilizadas as plataformas Git e Github para a hospedagem e criação do pacote. O Git é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, enquanto o GitHub é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou de código aberto de qualquer lugar do mundo.

Foi necessário a criação de repositórios no github para a hospedagem das funções e documentação do pacote criado. Para a primeira fase do projeto os códigos e documentação estão disponíveis em: [www.github.com/NomedoAutor/AHP](../../TPC02/Downloads/www.github.com/NomedoAutor/AHP). Já os da nova etapa estão disponíveis em: [www.github.com/NomedoAutor/AHP2](http://www.github.com/NomedoAutor/AHP2), neste está visível a seguinte arrumação de pastas:

* R: Códigos do pacote.
* Banco\_de\_dados: Dados em formato xlsx (objeto Excel) para testes e criação de base de dados.
* Data: Dados em formato rda (objeto R).
* Man: Documentação das funções e dados do pacote.
* Vignettes: Arquivos de ajuda à utilização do pacote em formato markdown.
* Documentacao: Documentos escritos do pacote

A estrutura citada foi criada utilizando o pacote *devtools*, que é uma biblioteca criada pelo Hadley Wickham que está disponível no github e também no CRAN. Esta biblioteca é bastante conhecida e utilizada pelos desenvolvedores de pacotes em R. Nela existem ferramentas que tornam o processo de criação de bibliotecas mais simples.

Pelo motivo deste pacote ter sido criado em um computador que possui sistema operacional Windows, foi necessário a instalação do *Rtools,* que é uma coleção de recursos para a criação de pacotes para *o* R no *Microsoft Windows* ou para a criação do próprio R, criado pelo Prof. Brian Ripley e Ducan Murdoch; atualmente mantido pela Jeroen Ooms.

Para a criação das funções do pacote AHP foram utilizadas as seguintes bibliotecas auxiliares*: formattable, dplyr, tibble, readxl e tidyr*. As funções foram criadas com o intuito de serem simples para a aplicação e apresentar uma saída visualmente intuitiva para o usuário

Implementou-se funções para facilitar ao usuário a aplicação do método. Para os cálculos envolvidos no método foram implementadas as funções com as seguintes etapas:

* Ler os dados (matrizes de julgamentos fornecida pelo usuário);
* Calcular os pesos e a consistência;
* Retornar tabela com os pesos finais de cada alternativa, informando o índice de consistência dos julgamentos de cada critério considerado no problema.

Foi utilizado a biblioteca *shiny* para a criação de um aplicativo que oriente o usuário a criar as matrizes paritárias de forma interativa no próprio ambiente R. O aplicativo ainda está com um visual e aplicações simples pois ainda está na fase piloto.

# Resultados e Discussão

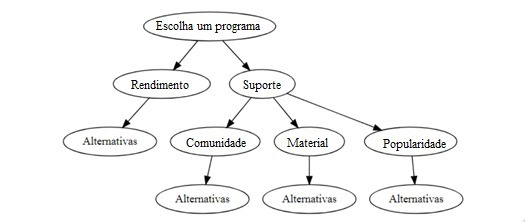
Na primeira fase do projeto cujo o pacote está disponível em [www.github.com/NomedoAutor/AHP](file:///F:/GitHub/AHP2/documentacao/www.github.com/NomedoAutor/AHP) foram criado códigos para a utilização do método AHP para problemas somente com um critério. Já a nova etapa que será apresentada neste trabalho, que está disponível [www.github.com/NomedoAutor/AHP2](../../TPC02/Downloads/www.github.com/NomedoAutor/AHP2), foram desenvolvidas funções que aplicam o método AHP para problemas com critérios e subcritérios, além de procedimentos auxiliares para facilitar a utilização do usuário.

Para realização do método, o programa espera que o usuário possua as matrizes de paridade em planilhas no formato xlsx, ou que o usuário utilize as funções auxiliares do pacote para criar as matrizes como objeto R.

Foram desenvolvidas algumas bases de dados exemplo que podem ser acessadas ao carregar o pacote, usando a função help do R em cada base é possível visualizar um cenário fictício em que um especialista criou as matrizes de julgamento e como estão estruturadas. Essas bases foram chamadas de BD1, BD2 e BD3. Os dados de BD1 e BD2 são provenientes de um problema somente com um nível de critério, isto é, sem subcritérios, já BD3 possui critérios e subcritérios.

Os códigos foram criados para aceitarem tanto objetos lista com matrizes de paridade no R e também planilhas em formato xlsx. As arrumações das matrizes devem respeitar a seguinte posição: Primeiro a matriz de comparação dos critérios em relação ao objetivo, em seguida, se houver subcritério a matriz de comparação dos subcritérios em relação ao seu critério, e pôr fim a matriz de comparação das alternativas mediante a cada subcritério (se houver) ou a cada critério.

Para exemplificação observe a figura 1, que apresenta uma hierarquia para um problema cujo o objetivo é escolha de uma linguagem de computação



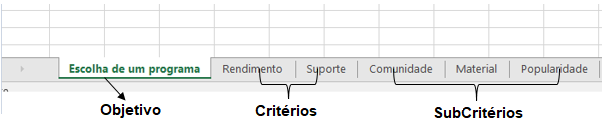
**Figura 1 –** Hierarquia para escolha de uma linguagem de computação

Fonte: Autor, 2019

É necessário que as matrizes de paridade estejam organizadas do seguinte modo:

1. Matriz de comparação dos critérios a luz do objetivo que é a escolha de uma linguagem.
2. Matriz de comparação das alternativas mediante ao critério rendimento (não possui subcritérios).
3. Matriz de comparação dos subcritérios Suporte.
4. Matriz de comparação das alternativas mediante ao subcritério Comunidade.
5. Matriz de comparação das alternativas mediante ao subcritério Material.
6. Matriz de comparação das alternativas mediante ao subcritério Popularidade.

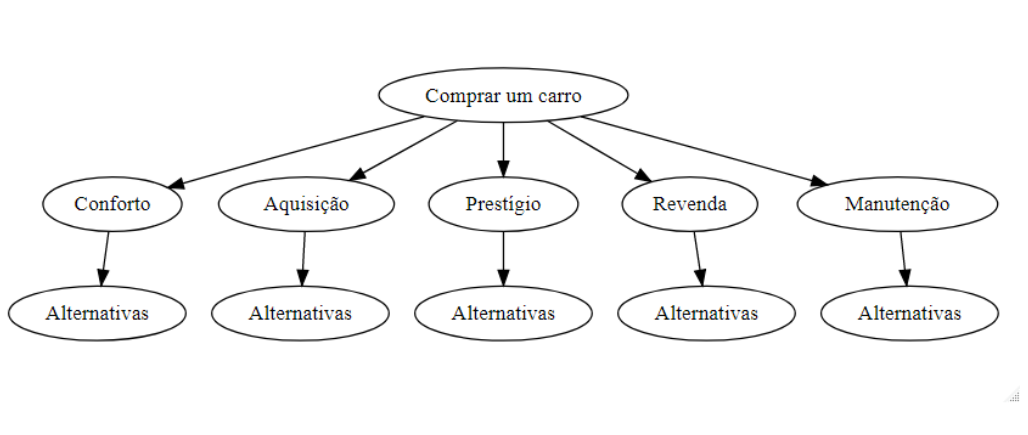
Caso o usuário escolha usar um arquivo de planilhas em formato xlsx, as planilhas devem ser organizadas de acordo com a figura 2.



**Figura 2 –** Arrumação de planilhas xlsx

Fonte: Autor, 2019

Atualmente o pacote conta com 6 funções e essas são: ahp\_geral(), ranque(), CR(), matriz\_julgamento(), calcula\_prioridade(), formata\_tabela(). Para exemplificação serão utilizadas a base de dados BD1 já disponível no pacote que apresente um cenário fictício de escolha de um carro entre 3 alternativas como também um segundo exemplo representado pela hierarquia apresentada na figura 1. O cenário do BD1 é um problema de apenas um nível de critérios sem subcritérios, estes critérios são: Conforto (CF); Aquisição (AQ); Prestígio (PS); Revenda (RV) e Manutenção (MA). A hierarquia pode ser observada na figura 3.



**Figura 3 –** Hierarquia do problema do BD1

Fonte: Pacote AHP, 2019

**Conclusão**

A escolha da linguagem R para implementação e manutenção do pacote tem se mostrado viável, consistente e segura. Foram implementadas novas funções, ampliando o alcance do método para problemas mais gerais, contemplando níveis de critérios e subcritérios. O pacote foi elaborado para facilitar a entrada dos dados, assim dando ao usuário possibilidades de escolha, podendo ser utilizado o próprio R, ou planilhas externas, com os dados estruturados de acordo com os critérios considerados na hierarquização do problema. As funções foram implementadas na linguagem R de forma a produzir uma tabela final com um resumo das probabilidades de cada critério, subcritérios e alternativas. Foram desenvolvidas opções de formatação da tabela final de resultados, ampliando sua aplicabilidade para uma rápida interpretação por parte do tomador de decisão, podendo ser publicada em apresentações ou artigos.

A principal vantagem deste pacote é a facilidade que o usuário tem para realizar a entrada de dados que também foi desenhada para ser feita por arquivo do tipo xlsx. As tabelas de saída são apresentadas de forma compacta para que o tomador de decisão possa atingir seu objetivo com rapidez e eficiência.

O pacote apresenta tutorial para que o usuário consiga utilizá-lo baseando-se num exemplo prático, tornando-se útil para gestores com conhecimentos básicos de linguagem de programação.

Como ações futuras será aprimorada a aplicação *shiny* com foco na utilização das funções do pacote AHP para usuários que não dominam o R, para isso o aplicativo será transformado em um *addin* para o RStúdio.

Este projeto está sendo desenvolvido na Universidade Federal Fluminense com o apoio da bolsa do programa de Iniciação Científica (PIBIC).

# Referências

R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL [http://www.R-project.org/,](http://www.r-project.org/) 2018.

WICKHAM, H.; BRYAN, J. *readxl: Read Excel Files. R package version 1.0.0.* <https://CRAN.R-project.org/package=readxl>, 2017.

MÜLLER, K. & WICKHAM, H. *tibble: Simple Data Frames. R package version 1.4.2.* <https://CRAN.R-project.org/package=tibble>, 2018.

WHICKHAM, H. & HENRY, L. *tidyr: Tidy Messy Data. R package version* 1.0.0. <https://CRAN.R-project.org/package=tidyr>, 2019.

WHICKHAM, H.; HENRY, L.; FRANÇOIS, L. dplyr: *A Grammar of Data Manipulation. R package version* 0.8.3. <https://CRAN.R-project.org/package=dplyr>, 2019.

Toth, Paolo & Vigo, Daniele. *Vehicle routing : problems, methods, and applications.* Bologna-Itália: University of Bologna, 2Ed, ISBN 978-1-611973-58-7, 2002.

GOMEDE, E. & MIRANDA, R. Utilizando o Método Analytic Hierarchy Process (AHP) para Priorização de Serviços de TI: Um Estudo de Caso, 2012. URL: <http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2012/0041.pdf>

RIBEIRO, M. C. C. & ALVES, A. S. Aplicação do método AHP com a mensuração absoluta num problema de seleção qualitativa. Revistasg, 2016. URL: <http://www.revistasg.uff.br/index.php/sg/article/view/988/493>

**Anexo 1: Códigos auxiliares para aplicação do RouteR.**