

## Uma Proposta de Solução para o

### Problema da Localização de Instalações:

#### Uma Abordagem Baseada em Técnicas de Aprendizado de Máquina

**Bárbara de Oliveira Andrade, Prof. Dr. Arnaldo Rabello Aguiar Vallim Filho**

Sistemas de Informação - Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie (Mackenzie) – São Paulo – SP – Brasil

31687059@mackenzista.com.br, arnaldo.vallim@mackenzie.br

**Abstract.** This final paper aims to identify logistical problems in locating product distribution facilities and using data science, artificial intelligence/machine learning tools to propose solutions using classical metaheuristics to obtain solutions that optimize safety, accessibility, minimizing the overall cost of operation and maximizing the distribution center's efficiency.

The study was conducted under the guidance of Prof. Dr. Arnaldo Rabello Aguiar Vallim Filho and has scientific research on the subject as support material, with the aim of having the best possible solution within the proposed subject.

**Resumo.** Esse trabalho de conclusão de curso realizado, tem como objetivo identificar problemas logísticos de localização de instalações de distribuição de produtos e através do uso de ferramentas de ciência de dados, inteligência artificial/machine learning propor soluções utilizando metaheurísticas clássicas a fim de obter soluções que otimizem as frentes de segurança, acessibilidade, minimizando o custo global de operação e maximizando a eficiência do centro de distribuição.

O estudo foi conduzido com a orientação do Prof. Dr. Arnaldo Rabello Aguiar Vallim Filho e tem como material de apoio pesquisas científicas sobre o tema, com o intuito de termos a melhor solução possível dentro do tema proposto.

### 1. Introdução

A definição de um centro de distribuição (CD) é de grande importância na cadeia de suprimentos, pois interfere diretamente nos prazos e custos de entregas para os clientes e empresas.

Para auxiliar na definição da localização dos CD's, a proposta deste estudo é utilizar dos recursos de ciência de dados para propor um modelo que otimize a distribuição dos centros levando em consideração variáveis como demanda, segurança, acessibilidade e custos envolvidos.

#### 1.1. Contextualização e Relevância do Tema

Em 2020 o mundo vivenciou o início da pandemia de COVID-19, que gerou grandes impactos na economia global e no comportamento da sociedade, dada a necessidade do distanciamento social. Uma alteração economicamente relevante foi o comportamento de compras, onde houve a priorização das compras online em relação as compras presenciais. Segundo o SEBRAE (2020), as vendas online de supermercados tiveram um aumento de 16%.

Com o aumento das vendas online houve também alterações no fluxo logístico, pois o produto que antes saía de um centro de distribuição para uma loja onde era comprado pelo cliente, hoje é demandado pelo cliente sem que ele retire o produto dessas lojas, optando pelo recebimento em sua residência, por exemplo. De acordo com o Correio Braziliense (2020), [...]com o isolamento social e as medidas restritivas para evitar a disseminação do novo coronavírus, a alternativa dos consumidores foi recorrer ao e-commerce. Mas desafios têm sido enfrentados pelos usuários: assim como as compras on-line no Brasil aumentaram durante o período de pandemia, a demora no recebimento de produtos adquiridos pela internet também tem sido recorrente. [...] E uma consequência dessa alteração é justamente o fato de que os locais de entregas designados pelos clientes podem estar localizados em locais muito próximos ou muito distantes dos centros de distribuição das empresas, o que pode impactar nos custos e prazos de entrega.

Tem-se como definição que [...]o centro de distribuição - diferente do armazém geral – tem como finalidade gerenciar o fluxo de produtos e informações associadas, de modo que possa contribuir para a redução das distâncias, diminuindo os prazos de entrega, contribuindo para o atendimento das necessidades dos consumidores [...] (SILVA, A. 2015).

Levando em conta que se vive em uma época em que os dados estão por toda a parte, desde a quantidade de horas que se utiliza o celular até quantas vezes se passa em um determinado local em um determinado período no tempo. Os dados são considerados por muitos como o “petróleo” do momento, segundo o secretário geral da ONU, Guterres (ONU NEWS, 2020) os “[...] dados atuais, seguros, oportunos e confiáveis ajudam a entender o mundo em mudança e a impulsionar as transformações necessárias, sem deixar ninguém para trás.”

Um exemplo de uso de dados é o aplicativo *Waze*, que utiliza dados diversos para a determinação de rotas para os usuários entre uma localização origem e uma dada localização destino.

Nesse contexto, o papel assumido pela Ciência de Dados se torna vital para a compreensão de cenários possíveis, auxiliando na construção de modelos preditivos para dar insumos para a tomada de decisão nas mais diferentes áreas do conhecimento, e não seria diferente na definição da solução ótima no quesito localização de centros de distribuição para otimizar os custos e prazos para o cliente e as empresas.

## **1.2. Objeto de Pesquisa**

### **Problema de Pesquisa**

Este problema busca a definição da quantidade e localização de um conjunto de Centros de Distribuição de Carga (CD).

Os CD's deverão ser abastecidos com carga, por fontes de abastecimento (fornecedores) com capacidade de atendimento e localização conhecidas. Os CD's deverão atender pontos (clientes) distribuídos geograficamente em uma região, com demanda e localização conhecidas.

O que se busca é a solução ótima desse problema, que defina a quantidade e localização dos CD's.

Este problema poderia ser resolvido com o apoio de técnicas de Inteligência Artificial / Aprendizado de máquina (*Machine Learning*) que são muito utilizadas para

diversos tipos de problemas em Ciência de Dados, em diferentes áreas de aplicação. Desta forma, poderia ser definida a seguinte pergunta de pesquisa:

“Como seria o desempenho de técnicas de Inteligência Artificial, aplicadas na solução de um problema da localização de instalações? Como poderia ser implementado um processo de solução do problema com o apoio desse tipo de técnica, e como pode ser implementada uma visualização gráfica da solução ótima que seria encontrada?”

### **Hipótese**

Para a definição da solução ótima, considera-se que a quantidade e localização dos CD's seria estabelecida em função das seguintes hipóteses:

i) Fator decisivo para a identificação da solução ótima seria a distribuição geográfica dos pontos de demanda na região de estudo e do valor da demanda em cada ponto;

ii) O nível de segurança dos locais candidatos à instalação de um CD seria fator direcionador do processo decisório;

iii) A acessibilidade dos locais candidatos à instalação de um CD seria outro fator que direcionaria a decisão de melhores locais;

iv) O custo envolvido na operação deverá direcionar a solução, de forma que a solução ótima seja aquela de menor custo global da operação;

v) Os principais componentes de custo que devem direcionar a solução ótima seriam: custo do produto ofertado em cada fonte de abastecimento, custo do transporte entre fonte de abastecimento e CD, custo do transporte entre CD e ponto de demanda e custo do local em que será instalado o CD (custo da terra).

## **1.3. Objetivos do Estudo**

### **Objetivo Geral**

O objetivo geral desta pesquisa problema é propor um processo de modelagem para solução ótima do problema logístico da localização de instalações, definindo a quantidade e localização de um conjunto de Centros de Distribuição de Carga (CD) para atender pontos de demanda distribuídos geograficamente em uma região.

A solução deverá considerar restrições operacionais envolvendo as fontes de abastecimento de carga, com capacidade de atendimento e localização conhecidas e as demandas a atender.

### **Objetivos Específicos**

Em termos de objetivos específicos tem-se:

- Estudar uma combinação de técnicas para solução do problema;
- Desenvolver experimentos dos modelos propostos usando bases de dados públicas;
- Implementar os algoritmos de solução em ferramentas de Aprendizado de Máquina, como a linguagem R;
- Estudar a possibilidade de uma visualização dos resultados com o apoio de bibliotecas do R.

## **1.4. Justificativa**

O problema logístico de localização de instalações é relevante por conta do impacto gerado nas entregas de produtos, fato esse que foi muito evidenciado por conta da pandemia do Coronavírus. Com a crescente das compras através de serviços online, ficaram evidentes alguns problemas de logística, tais como, o atraso nas entregas, aumento no custo de operação, entre outros já eram problemas que aconteciam em situações normais e com a pandemia esses problemas ficaram ainda mais explícitos, gerando também a oportunidade de os consertar.

### **1.5. Delimitação do Estudo**

A questão da delimitação deste estudo foi analisada em relação aos pontos descritos a seguir:

- ✓ Delimitação Organizacional: *Não se vislumbra que haja este tipo de delimitação. Entende-se que o pode se aplicar o estudo a praticamente qualquer tipo de organização*
- ✓ Delimitação Geográfica: *Também não se vislumbra que haja este tipo de delimitação. Os dados são genéricos, podendo ser adaptados a qualquer localidade.*
- ✓ Delimitação Temporal: *A exemplo dos itens anteriores, aqui também não se vislumbra que haja este tipo de delimitação. Os dados são genéricos, podendo ser adaptados a qualquer período no tempo.*
- ✓ Delimitação por Indicador(es) de Desempenho: *São utilizados indicadores de Custo de operação, Aumento de produtividade e redução de lead time de entrega.*
- ✓ Delimitação por Categoria(s) de Profissionais; *O estudo é voltado para operações logísticas, assim, considera-se que teria como usuários os profissionais da área de logística.*

### **1.6. Organização do Estudo**

Nas próximas seções será apresentado o detalhamento dos textos que foram selecionados como referencial teórico, bem como a explicação das metodologias de pesquisa que serão utilizadas ao longo de todo desenvolvimento do TCC, e finalizando o documento, serão descritas as referências bibliográficas que foram utilizadas.

De forma mais detalhada, além desta seção 1, de Introdução, este Trabalho de Conclusão de Curso (TCC) está organizado conforme as seções descritas na sequência.

Seção 2 – Referencial teórico que será utilizado no desenvolvimento do TCC

Seção 3 – Metodologia que será utilizada para composição de toda a pesquisa

Seção 4 – Desenvolvimento e Resultados onde será explorado o modelo desenvolvido

Seção 5 – Conclusões e Recomendações, que servirá para demonstrar qual a conclusão obtida a partir do modelo proposto e quais suas possíveis aplicações.

## **2. Referencial Teórico**

### **2.1. Base Conceitual**

#### **2.1.1 Problemas de localização de instalações**

Brandeau e Chiu (1989) apresentam o histórico das formulações de problemas de instalações, que se iniciou em 1909 por Alfred Weber que considerou o problema de escolher a localização de um único armazém a fim de minimizar a distância total entre ele e os clientes.

Desde então, vários problemas foram propostos ao longo do tempo, como veículos do corpo de bombeiros por Valinsky em 1955 e de forma mais teórica e geral, Hakimi em 1964 considerou o problema de identificar a melhor localização para uma ou mais instalações a fim de minimizar as somas das distâncias ou a máxima distância entre as instalações e os pontos em uma rede (Brandeau e Chiu, 1989).

Eles também apresentam a definição de um problema de localização como:

“[...] is a spatial resource allocation problem. In the general location paradigm, one or more service facilities ("servers") serve a spatially distributed set of demands ("customers"). The spatial topology being modeled may be a general network, or a specialized network (e.g., a tree). The objective is to locate facilities (and perhaps allocate customers to servers) to optimize an explicit or implicit spatially dependent objective.” (Brandeau e Chiu, 1989).

Ou seja, trata-se de um problema de **alocação espacial** composto por instalações distribuídas em uma área que devem atender a demandas, cujo objetivo é alocar as instalações a fim de otimizar um critério, seja minimizando custos ou distâncias ou maximizando a eficiência.

Os exemplos de problemas de localização típicos, segundo Brandeau e Chiu (1989), são: **Network Design** (Desenho de redes), que trata do desenho de um sistema de tratamento de água ou esgoto a fim de minimizar o encanamento, por exemplo; **Warehouse Location** (Localização de armazém), que trata da localização de armazéns a fim de minimizar o custo total do atendimento em uma região que serve um número de clientes; **Fire Box Coverage** (Cobertura de atendimento dos bombeiros, em tradução livre), que aloca um determinado número de instalações a fim de minimizar a distância máxima necessária para chegar até qualquer cidadão e; **Competitive Facility Location** (Localização de instalações em ambiente competitivo, em tradução livre), que visa a introdução de instalações em uma região já atendida por concorrentes cujo objetivo é maximizar os lucros ou o *Market share*.

### 2.1.2 Introdução aos Modelos de localização (*Facility Location*)

O modelo de localização pode ser utilizado para identificar soluções para diversos tipos de problemas, não só no processo logístico em si, mas em vários assuntos. Segundo Daskin (2008):

[...] Applications of location modeling include locating emergency medical service (EMS) bases, fire stations, schools, hospitals, reserves for endangered species, airline hubs, waste disposal sites, and warehouses to list only a small subset of the numerous areas in which location models have been applied. Location models have also found applications in nontraditional areas, including medical diagnosis, vehicle routing, alignment of candidates and parties along a political spectrum, and the analysis of archeological sites. [...]

Segundo ReVelle et al. (2008), os modelos de localização, embora possam tratar de situações diferentes possuem algumas variáveis em comum como um espaço, métrica,

clientes e instalações cuja localização deve ser definida de acordo com alguma função objetivo:

[...] Even though the contexts in which these models are situated may differ, their main features are always the same: a space including a metric, customers whose locations in the given space are known, and facilities whose locations must be determined according to some objective function. [...]

### 2.1.3 Uso de clusterização em problemas de localização

A *clusterização* tem por objetivo identificar padrões para segregar os dados em agrupamentos, nas palavras de Vale (2006): [...] A análise de agrupamentos pode ser definida como o processo de determinação de  $k$  grupos em um conjunto de dados[...]. O que, segundo Barreto et al. (2007) [...] Recognizing groups of customers can be a good start to obtain good LRP solutions. [...], em tradução livre, podemos dizer que: reconhecer grupos de cliente pode ser um bom começo para obter boas soluções de problemas de localização (LRP).

### 2.1.4 Modelos Exatos

Dentre os modelos de localização existentes, ReVelle et al. (2008) destacam quatro: modelo analítico (*analytic models*), modelo contínuo (*continuous models*), modelo de rede (*network models*) e discreto (*discrete location*).

**Modelo analítico (Analytic models):** O modelo analítico é mais simples, com suposições de valores fixos ou uniformemente distribuídos. Para este modelo são utilizadas as seguintes variáveis: densidade da demanda  $p$ , tamanho da região de serviço  $\alpha$ , custo do local de instalação  $f$ , custo de transporte unitário por unidade de distância  $c$  e número de instalações  $n$ . Esse modelo é expresso por três funções, onde o custo total associado é:  $fn + \frac{pca}{3} \sqrt{\frac{2a}{n}}$ , função que é minimizada por  $n = \left(\frac{cp\sqrt{2}}{6f}\right)^{\frac{2}{3}}$ , em que o custo total é  $1.1447 Af^{\frac{1}{3}}(cp)^{\frac{2}{3}}$ . (ReVelle et al., 2008)

**Modelo contínuo (Continuous models):** Segundo ReVelle et al. (2008), [...] typically assume that facilities can be located anywhere in the service area, while demands are often taken as being at discrete locations [...], ou seja, em tradução livre: as localizações podem estar em qualquer local na área de serviço, enquanto a demanda pode ser considerada em localização discreta. Ainda segundo os autores, um problema clássico para esse modelo é o de rede (Weber problem), onde há uma única instalação que serve uma demanda  $m$  com coordenadas  $(x_i, y_i)$  com  $i = 1, \dots, m$  e a demanda (peso)  $w_i, i = 1, \dots, m$ . Onde o problema é encontrar a posição  $(x_0, y_0)$  para minimizar o custo, geralmente em linha reta utilizando a distância euclidiana. (ReVelle et al., 2008).

**Modelo de Rede (Network models):** O problema de rede é composto por conexões (*links*) e nós (*nodes*), sendo a demanda tipicamente originada pelos nós (ReVelle et al., 2008) e as instalações podem estar localizadas em qualquer lugar na rede (Daskin, 2008).

Ainda segundo Daskin (2008), o problema de instalações no modelo de rede é tratado como uma árvore onde a solução pode ser expressa pela complexidade de algoritmos  $O(n)$  onde  $n$  é o número de nós da árvore.

**Modelo Localização Discreta (*Discrete location*):** O modelo assume que dois conjuntos discretos, o de demanda (I) e o de Instalações (J) (ReVelle et al., 2008).

Nesse modelo a distância pode ou não ser representada pelo sistema métrico, assim como os custos podem ser definidos arbitrariamente (Daskin, 2008).

Daskin (2008), continua tipificando o modelo discreto em três grandes grupos: *Covering-based Models*, *Median-based Models* e *P-Dispersion*, em tradução livre seriam modelos baseados em cobertura, modelos baseados em medianas e P-dispersões.

Os *Covering-based Models* são compostos pelos seguintes subitens: definição de cobertura cujo desafio é minimizar as instalações necessárias para cobrir as demandas; maximização da cobertura que com base nas instalações necessárias busca maximizar a capacidade de cobertura e o P-centro que visa minimizar a distância percorrida para cobrir as demandas.

Já os *Median-based Models* possuem o P-mediana que minimiza a distância média entre as demandas e as p instalações próximas e possuem a taxa fixa que visa minimizar os custos de transporte e instalações.

Por fim, o *P-Dispersion* que é aplicado aos outros modelos, os que não se encaixam nas categorias anteriores, visa minimizar a distância máxima entre qualquer par de instalações.

### 2.1.5 Metaheurísticas

As metaheurísticas são algoritmos feitos para criar soluções aproximadas para problemas difíceis sem que seja necessária a criação de uma solução específica e que comumente são inspirados na natureza (Boussaïd et al., 2013).

Massuti (2016), explica que os problemas de localização são caracterizados por terem várias soluções possíveis e por esse motivo as metaheurísticas são muito úteis porque mesmo que não tragam uma solução exata, elas apresentam soluções satisfatórias por, em sua execução, manter os resultados das interações anteriores e ir modificando e aprimorando-os conforme o acúmulo de interações ocorre.

Podemos trazer duas categorias de metaheurísticas: *single-solution based* ou em tradução livre: metaheurísticas baseadas em soluções únicas e *population-based metaheuristics*, ou em tradução livre: metaheurísticas baseadas em população (Boussaïd et al., 2013).

As metaheurísticas baseadas em soluções únicas são chamadas de métodos de trajetória pois começam com uma solução inicial e vão se distanciando dela criando uma trajetória. Dois tipos dessa categoria são o *Simulated annealing* (Têmpera simulada) que remete ao processo de manipulação de metais onde há a exposição a altas temperaturas e depois um resfriamento prolongado para poder modelá-los e o *Tabu Search* (Busca Tabu) que foi desenhado para gerenciar um algoritmo de busca local embutido. Sua principal característica é ser baseado nos mecanismos da memória humana, que diferentemente do *Simulating Annealing*, *proporciona um aprendizado com o seu passado* (Boussaïd et al., 2013).

Já as chamadas metaheurísticas baseadas em população possuem o campo biológico: *evolutionary computation* (Computação Evolucionária), como o *Genetic Algorithm* e o *swarm intelligence*, ou inteligência de enxame que é aquele que possui o

grupo de algoritmos como *Ant Colony* e *Bee Colony* que são algoritmos inspirados nas interações sociais dos enxames ao invés do comportamento individual das abelhas (Boussaïd et. al., 2013).

O *Genetic Algorithm* é o mais conhecido no campo da computação evolucionária. Ele se baseia na combinação dos cromossomos e é bem genérico, podendo ser adaptado para várias funcionalidades e com base em sua configuração demonstrar as mutações que ocorrem assim como na recombinação genética (Boussaïd et. al., 2013).

As interações mais exploradas pela inteligência de enxame é a capacidade de *self-organization* (organização própria) e *division of labour* (divisão de trabalho) (Karaboga, 2005).

A capacidade de organização própria é definida como um conjunto de mecanismos que estabelecem regras básicas de interação entre os componentes do sistema, sendo eles o *positive feedback* (*feedback* positivo), *negative feedback* (*feedback* negativo), *fluctuations* (flutuações) como comportamentos aleatórios e erros e o *mutually tolerant individuals* (tolerância mútua dos indivíduos) (Karaboga, 2005).

Já no campo de *swarm intelligence* temos o *Ant Colony Optimization* que foi inspirado no comportamento das colônias de formigas, que ao buscar por alimentos iniciam uma busca randomizada próximo à colônia e marcam as rotas com feromônios para indicar os locais favoráveis, sendo que os locais mais próximos ficam com um excesso de feromônio e atraem mais formigas. (Boussaïd et. al., 2013).

Os algoritmos *Bee Colony* são comumente inspirados nos seguintes comportamentos: *bee dance and communication* (dança das abelhas e comunicação), que estão relacionados a comunicação da localização de fontes de alimento; *mating and marriage process* (acasalamento e processo de casamento) que explora o processo de reprodução da colmeia; *bee foraging* (busca de alimentos das abelhas) que explora o processo de busca por alimento; *nest site selection* (escolha do local do ninho) que utiliza como base o processo de escolha de um novo ninho quando a colônia está sobrecarregada; *navigation* (navegação) que se baseia na capacidade de aprendizado das abelhas sobre direções e distâncias em suas viagens; e *task selection* (seleção de tarefas) baseada no comportamento das abelhas que realizam suas tarefas sem um comando centralizado (Boussaïd et al., 2013).

Artificial Bee Colony (ABC): O algoritmo *artificial bee colony* é uma metaheurística baseada em população que explora o comportamento de comunicação e busca por alimentos das colônias de abelhas com foco no recrutamento para uma fonte de néctar e o abandono de uma fonte que é útil para resolver problemas de tráfego e transporte (Karaboga e Akay, 2009).

Segundo Karaboga e Akay (2009), esse algoritmo é constituído por três partes principais: fontes de alimento, operárias que buscam alimento, e as não operárias. Essas partes funcionam da seguinte forma:

- Fontes de alimento (*food sources*): No processo de escolha de uma fonte de comida, suas características são avaliadas, tais como proximidade, riqueza de energia, sabor do néctar, facilidade de extração que são resumidas em uma única quantidade, como qualidade (Karaboga e Akay, 2009);



- Operárias (*employed foragers*): Essas abelhas são operárias que exploram uma fonte específica de alimento e elas retornam para a colmeia com as informações de lucratividade, direções e distâncias (Karaboga e Akay, 2009);
- Não operárias (*unemployed foragers*): Assim são chamadas as abelhas que buscam nova fonte de alimentos que podem ser tanto as exploradoras (Masutti, 2016), (*scouts*), que buscam novos locais randomicamente ou observadoras (Masutti, 2016) (*onlookers*) que tentam identificar os locais com base nas informações trazidas pelas operárias que buscam alimento (Karaboga e Akay, 2009);

Ainda segundo Karaboga e Akay (2009), depois da localização de novas fontes potenciais, existem 3 escolhas possíveis:

- Pode-se abandonar a fonte de alimento e virar uma seguidora; ou
- Pode-se dançar para divulgar a informação e recrutar para a fonte identificada; ou;
- Pode-se continuar a busca na fonte de alimento sem recrutamento.

No *artificial bee colony* (ABC), em tradução livre: “[...] a solução representa a localização da fonte de alimento, a qualidade do néctar representa a aptidão da solução. A inicialização do algoritmo é realizada com a geração de uma fonte de alimento e depois as abelhas realizam buscas nas proximidades por outras fontes de alimentos para identificar uma fonte melhor e as fontes inferiores são removidas e assim por diante. [...]” (Sahin et. al., 2021).

## 2.2. Trabalhos Correlatos

Nesta seção são apresentados os principais *papers*, dissertações e relatórios técnicos utilizados como base para a formulação desse artigo.

Um primeiro artigo importante já citado anteriormente é o de BRANDEAU e CHIU (1989), que apesar de ser antigo, os conceitos lá apresentados permanecem atuais. Os autores trazem uma conceituação abrangente do problema de localização, trazendo as principais características, definições e exemplos de problemas conforme citado acima. Outro artigo nessa linha é o DASKIN (2008) que apresenta uma visão geral dos modelos de localização e detalha alguns tipos como a modelagem de localização discreta, conforme citado nos conceitos apresentados, relacionando pesquisas recentes na área.

Em REVELLE et al. (2008) tem-se uma apresentação dos tipos de modelagem para problemas de localização dando ênfase à modelagem de localização discreta (*discrete location*), conforme citado na seção 2.1 fornece uma bibliografia comentada de artigos em dois ramos da teoria e modelagem de localização discreta.

Já em VALE, M. N. DO (2008), é realizada uma revisão detalhada sobre o processo de análise de agrupamentos com a definição do conceito de clusterização, conforme citado neste documento, e como ele pode ser utilizado para revelar como os dados estão estruturados e resultam em um melhor entendimento sobre o negócio. Também é realizado o desenvolvimento de um aplicativo que visa não apenas a atender as deficiências presentes na maioria das ferramentas com esse fim, mas também a auxiliar, de forma mais completa, todo o processo de análise dos grupos.

Os conceitos anteriores são unidos em BARRETO, S., et al. (2007) que, em seu artigo trata o problema de localização (LRP) com a abordagem de clusterização para determinar o conjunto de CDs instalados, bem como as rotas de distribuição (começando

e terminando no CD). Cujo objetivo é minimizar os custos de encaminhamento e localização. Todas as versões obtidas por diferentes procedimentos de agrupamento foram testadas em muitas instâncias (adaptadas de dados da literatura) e os resultados foram comparados a fim de se obter algumas orientações quanto à escolha de uma técnica de agrupamento adequada.

De forma mais abrangente, em BOUSSAÏD, et al. (2013), é apresentada a visão geral sobre as Metaheurísticas apresentando seu conceito e trazendo exemplos e suas categorias, como as *single-solution based* e *population-solution based* que foi utilizado como base para escolha da metaheurística utilizada nessa pesquisa, a *Artificial Bee Colony* (ABC), que é aprofundada no reporte técnico de KARABOGA (2005), que discorre sobre o conceito de inteligência de enxame (*swarm intelligence*) e o comportamento das colmeias de abelhas destacando a relevância para a busca de alimentos, reprodução e escolha de novas colmeias. Assim como em KARABOGA e AKAY (2009), onde se discorre sobre o algoritmo Artificial Bee Colony (ABC) Apresentando o detalhamento de seu funcionamento e rerepresentando sua base teórica que utiliza como inspiração o comportamento de busca por alimentos, como é possível visualizar na citação apresentada na base conceitual (seção 2.1). Também é realizada uma comparação com o Algoritmo Genético (GA) e outros *population-based* comparando os resultados obtidos mostrando que o desempenho do ABC é melhor ou semelhante ao de outros algoritmos baseados em população, com a vantagem de empregar menos parâmetros de controle.

Também são avaliados artigos e dissertações mais recentes que utilizam a ABC, como é possível observar em MASUTTI (2016), que se demonstra a aplicação da metaheurística na solução de problemas de rotas, apresentando como exemplo o problema do “caixeiro viajante”. Ele menciona como as metaheurísticas são apropriadas para encontrar uma solução satisfatória para um problema que possui várias soluções possíveis. E em SAHIN, et al. (2021) são propostos dois novos algoritmos Artificial Bee Colony (ABC) para software orientado a objetos. Aqui é retomada a base conceitual do algoritmo original, conforme citado acima, para sustentar a alteração baseada em arquivo, que é aprimorado, testado e comparados para investigar o efeito do arquivo.

### **3. Metodologia da Pesquisa**

No que tange a metodologia deste trabalho de conclusão de curso, a ideia inicial se deu pelo interesse em se resolver problemas de logística utilizando ferramentas de ciência de dados.

Foi realizada uma análise e comparação sobre referenciais teóricos para fundamentação do projeto de TCC.

A sistematização dos dados foi feita com apoio de ferramentas estatísticas, gerando indicadores como gráficos, médias e outras medidas estatísticas de forma a entender o cenário atual e histórico dos centros de distribuições.

Após essa análise de dados, será possível e necessário extrair conclusões.

Nos termos de classificação da pesquisa, foi seguida a classificação clássica de:

- Quanto à Natureza, esta é uma pesquisa Aplicada, já que sua aplicação prática é imediata.

- Quanto à Forma de Abordagem, a pesquisa pode ser considerada quantitativa, pois desenvolve técnicas quantitativas.
- Quanto aos Fins (suas finalidades), a pesquisa é metodológica, já que apresenta uma metodologia de trabalho. Se utilizando de métodos de ciência de dados, para propor uma solução real e viável. Sendo então uma pesquisa propositiva.
- Quanto aos Meios (recursos empregados na pesquisa), serão utilizados os seguintes recursos: Bibliografia, pesquisa de campo e estudos de casos.

Nessa etapa do estudo está sendo utilizada a linguagem R no *RStudio* para o desenvolvimento.

A estratégia adotada foi a divisão do problema em duas partes: escolha das instalações que atenderão determinados clientes e escolha dos fornecedores que atenderão determinadas instalações. Para a identificar a solução ótima que minimize os custos foi utilizada a metaheurística Artificial Bee Colony (ABC), que foi escolhida a partir de pesquisa sobre metaheurísticas clássicas no ramo de computação evolucionária (*Evolutionary Computation*) que possuem técnicas mais utilizadas para populações, contemplando os algoritmos genéticos (*Genetic Algorithm*) e de Inteligência de Enxame (*Swarm Intelligence*), categoria a qual o ABC pertence. E para definição dos custos nos baseamos no cálculo de distância euclidiana e custos de fretes e penalidades que deverão ser acrescidas aos custos quando a demanda de um cliente específica for atendida parcialmente por um determinado fornecedor ou quando a demanda de um cliente não foi atendida integralmente pelo fornecedor determinado.

Em termos de sequência do desenvolvimento, o estudo procurou seguir as etapas descritas abaixo:

- **Levantamento Bibliográfico**, onde foram procurados artigos e dissertações de autores tradicionais nas áreas de problemas de localização e metaheurísticas, como DASKIN (2008) e KARABOGA (2005). Assim, como artigos mais recentes e que demonstram a aplicação da Artificial Bee Colony em problemas de localização, como MASUTTI (2016) e SAHIN, et al. (2021).
- **Levantamento de Bases de Dados**: para esse ponto foi realizada a construção de uma base personalizada com itens hipotéticos que se adaptasse a estrutura de dados necessária à execução do modelo. Para tanto, foram criadas duas bases externas ao código, a base de clientes/instalações candidatas, que possui 110 registros (100 clientes e 10 instalações) e a base com a lista de fornecedores possíveis com 20 registros.

As bases seguem a seguinte estrutura de parâmetros de entrada: Código da instalação/cliente, que serve como título para os itens que serão avaliados; Coordenada X e Coordenada Y, que demonstram a localização cartesiana que é utilizada para o cálculo da distância; demanda/capacidade, que no caso dos clientes guardam a demanda de atendimento, no caso das instalações demonstra a capacidade de atendimento e no caso dos fornecedores demonstra a capacidade de oferta para as instalações; Penalidade por problemas de acessibilidade, que acata um índice de 1 a 10, onde 1 representa poucos problemas e 10 o nível máximo; e por fim, Penalidades por problemas de Segurança, que também acata um índice de 1 a 10, onde 1 representa o menor índice de problemas, enquanto 10 representa o maior.

- **Definição de Técnicas de Abordagem do Problema**: Dentro do problema de localização de instalações, foram observadas as aplicações de metaheurísticas para

propor soluções para esses problemas. Portanto, foram pesquisadas técnicas compatíveis com o trabalho, como as metaheurísticas baseadas em população, cujas opções estavam dentro do espectro de computação evolucionária, que contempla algoritmos genéticos e inteligência de enxame. Dentro desse público foi escolhida a inteligência de enxame, que se baseia no comportamento de insetos e nesse público a escolha realizada foi pela Artificial Bee Colony.

- **Escolha das Ferramentas Computacionais:** Para desenvolvimento do modelo foi escolhida a linguagem R, que é tradicional para desenvolvimento de modelos estatísticos e amplamente utilizada em ambiente acadêmico na área de Ciência de dados.
- **Desenvolvimento de Algoritmos:** para esse ponto, tendo como base a metaheurística ABC, a abordagem escolhida foi de separação do algoritmo em fases, sendo que na primeira fase seria descrita a função objetivo da escolha de instalação por clientes e a segunda fase, que descreveria a função objetivo da escolha dos fornecedores que atenderiam as instalações definidas na primeira fase.

Outro ponto que foi avaliado foi a interação das variáveis de entrada que precisavam ser consolidadas para que um único retorno fosse realizado para a ABC, nesse caso foi o Custo global, que contempla os inputs de fretes e penalidades.

- **Desenvolvimento de Experimentos:** Para os experimentos foram utilizadas as bases populadas com itens hipotéticos que foram formulados para essa pesquisa. As mesmas bases foram processadas 35 vezes de forma sequencial em um curto espaço de tempo e tiveram seus conjuntos de arquivos de saída devidamente armazenados em pastas numeradas conforme sua ordem de execução.
- **Análise de Resultados:** os arquivos gerados a partir da execução dos 35 testes foram consolidados de forma a listar os custos globais de todas as execuções, que em média geram 1.800 resultados por execução, sendo que foram comparadas os 1<sup>os</sup> resultados de cada uma das 35 execuções com os últimos resultados delas, que representam a solução ótima. Esse público foi comparado para as saídas da fase 1 e da fase 2 e esses pontos foram plotados em gráficos para avaliar a amplitude e verificar se houve aumento ou redução dos custos entre os pontos.
- **Documentação do Estudo:** ela foi construída paralelamente a execução da pesquisa, sendo elencados primeiramente a introdução, o objetivo, a hipótese e o referencial teórico. Posteriormente foi incluída a metodologia e por fim o desenvolvimento e a conclusão da pesquisa.

#### 4. Desenvolvimento do Estudo e Resultados

Um primeiro ponto a ser definido no desenvolvimento do estudo foi a seleção de uma base de dados para os experimentos que seriam necessários para teste dos algoritmos que seriam construídos. Assim, neste estudo foi utilizada uma base de dados proposta por Christofides et al. (1979), que tem sido muito utilizada na literatura para testes de algoritmos, que é composta por 13 conjuntos de pontos, variando entre 13 e 101 pontos. Os conjuntos de dados da base não representam casos reais concretos, mas procuram retratar o que em geral ocorre em operações logísticas. Nesta pesquisa foi utilizada a base com 101 pontos. A base foi obtida no repositório “*The vehicle routing problem repository*” (VRP-REP, 2021), que é uma plataforma aberta de compartilhamento de dados de instâncias de problemas de roteirização de veículos, para utilização em estudos

de *benchmarking* de técnicas de soluções para este tipo de problema e de outros problemas similares.

Em relação à ferramenta computacional para construção e testes dos algoritmos, foi utilizada a linguagem R (CRAN,2021) em ambiente do RStudio (RStudio, 2021). O R é uma ferramenta de código aberto (*open source*), extensamente utilizada, que junto com o Python, se constituem nos dois principais instrumentos computacionais usados em Ciência de Dados na área de Ciência da Computação. O R tem um conjunto de bibliotecas com a implementação de algoritmos de aprendizado de máquina e de técnicas matemáticas e estatísticas dos mais diferentes tipos, inclusive para otimização de operações, o que facilitou o desenvolvimento dos algoritmos para este estudo. O RStudio é um ambiente de desenvolvimento integrado (*IDE - Integrated Development Environment*) para R e está disponível em código aberto e em edições comerciais, sendo executado em desktop. Nesta pesquisa, foi utilizada a versão de código aberto.

Uma vez definidas base de dados e ferramentas computacionais, o passo seguinte foi o desenvolvimento dos algoritmos. O framework desenvolvido tem duas Fases Principais (dois algoritmos) e uma Fase Intermediária, que ocorre entre as duas principais, conforme descritas abaixo:

FASE 1: Algoritmo de definição das Instalações por Cliente

FASE INTERMEDIÁRIA:

Etapa 1. Processo de Extração de Dados da Fase 1 e Criação de Repositório para Fase 2

Etapa 2. Processamento de Cálculos de Variáveis para Fase 2

Etapa 3. Criação de Base de Dados com Variáveis Calculadas para Fase 2

FASE 2: Algoritmo de definição de Fornecedores por Instalação

A fase 1 e a modelagem de dados foram aproveitadas de outros projetos já desenvolvidos, e neste projeto foram desenvolvidas a fase Intermediária e fase 2.

A fase 1 utiliza uma biblioteca do R que implementa a metaheurística ABC (*Artificial Bee Colony*), que interage com os dados por meio de uma função objetivo, que minimiza os custos da operação. O processo roda por um determinado número de iterações, identificando a configuração de pontos alocados a cada instalação logística e já definindo as rotas de atendimento desses pontos, partindo da instalação logística que os atende e retornando a essa mesma instalação. A biblioteca utilizada denominada “MetaOpt”, (Riza, 2019), tem em uma de suas funções a metaheurística ABC implementada. Esta função tem uma particularidade que exige que a entrada de dados seja feita em um único vetor. Assim, o problema deve passar por uma modelagem de dados que contemple essa exigência. Essa modelagem de dados é descrita na próxima seção.

#### **4.1. Modelagem dos dados: Representação de Rotas e Instalações em um único Vetor**

A proposta de modelagem contempla em um mesmo vetor os pontos de demanda de uma região a ser atendida e os números das instalações que atendem esses pontos.

Assim, tendo-se "n" Pontos e "m" Instalações, o Vetor terá "n + m" posições:

- os Pontos são numerados de “1” a “n”;

- as Instalações são numerados de “n+1” a “m”;

### Representação das Rotas de Atendimento dos Pontos de Demanda:

Considerando-se um vetor com "n + m" posições, tendo como conteúdo das suas posições uma distribuição aleatória de valores inteiros entre 1 e “n + m”, define-se que: os pontos alocados a um veículo (uma rota), são aqueles que estão no Vetor entre a posição em que está o número desse Veículo e a posição em que aparece o número da próxima instalação.

#### Observação:

1. Quando a sequência de números de Pontos após o número de uma Instalação atinge o final do vetor, considera-se que os Pontos alocados àquela Instalação (àquela rota) são esses, dessa sequência, mais todos do início do vetor, até que se atinja o primeiro número de uma Instalação (faz-se um retorno ao início do vetor).
2. Quando se tem dois números de Instalações em seguida, considera-se que a primeira dessas Instalações não recebeu a alocação de nenhum ponto. A instalação foi desativada.

Na sequência tem-se um exemplo: sejam 10 pontos e 3 instalações → tamanho do vetor = 13 posições:

**Tabela 1 - Estrutura de Vetor a ser Permutado por uma Metaheurística (MH) - ILUSTRAÇÃO**

PONTOS										Instalações		
1	2	3	4	5	6	7	8	9	10	11	12	13

Neste exemplo tem-se:

- Pontos são representados por números inteiros entre 1 e 10;
- Instalações são representadas por números inteiros entre 11 e 13;

O conteúdo de cada posição do vetor irá corresponder à sequência de pontos e instalações seja o vetor com as posições rearranjadas, que foi gerado por uma metaheurística: vetor x = c (3,2,4,12,10,7,9,8,11,5,6,13,1).

**Tabela 2 - Representação esquemática do vetor “x”**

PONTOS										Instalações		
1	2	3	4	5	6	7	8	9	10	11	12	13
3	2	4	12	10	7	9	8	11	5	6	13	1

O vetor acima indica a ordem de cada ponto/instalação, sendo que o ponto 3 ocupa a 1ª posição, ponto 2 a 2ª, o ponto 1 a 13ª, etc.

Note que a instalação nº. 11, está na posição 9 do vetor (é o 9º na sequência), a nº. 12, está na posição 4 do vetor (é o 4º na sequência) e a instalação nº. 13, está na posição 12 do vetor (é o 12º na sequência).

O que se busca, agora, é dado esse “vetor permutado”, que foi gerado por uma metaheurística, identificar nesse vetor as sequências de pontos alocados a cada instalação (as rotas).

No caso do exemplo, têm-se as seguintes alocações de pontos às instalações:

- Instalação 11: atende os pontos 5 e 6, que estão no vetor entre o valor 11 e o valor 13;

- Instalação 12: atende os pontos 10, 7, 9 e 8, que estão no vetor entre o valor 12 e o valor 11;
- Instalação 13: atende os pontos 1, 3, 2 e 4. O ponto 1 está no vetor após o valor 13 e os demais estão no início do vetor, antes de aparecer o valor 12.

Tem-se assim, 3 rotas: rota da instalação nº. 11, que contempla os pontos 11, 5, 6, 11; a rota da instalação nº. 12, com os pontos 12, 10, 7, 9, 8, 12; e a rota da instalação nº. 13 que contém os pontos 13, 1, 3, 2, 4, 13. Faz se observar que todas as rotas têm formato de “Gotas” (iniciam e terminam no mesmo ponto).

#### **4.2. FASE 1: Algoritmo de definição das Instalações por Cliente**

O objetivo do algoritmo é receber um vetor “x” rearranjado, gerado por uma metaheurística; traduzir esse vetor em um conjunto de rotas; aplicar a função objetivo ( $fO$ ) a essa configuração de rotas, e retornar para a metaheurística o valor da  $fO$  calculada.

Etapas do algoritmo:

00. Recebe vetor “x” da metaheurística e transforma em vetor permutado;
01. Localiza a posição das instalações no vetor permutado;
02. Identifica ordenação das instalações no vetor permutado;
03. Cria um vetor de rotas;
04. Identifica posição de início de cada rota no vetor de rotas;
05. Identifica nº. de pontos em cada rota;
06. Cria rotas em "gotas" (rotas circulares);
07. Identifica início/fim de cada rota no vetor de rotas em gotas;
08. Função objetivo: calcula distâncias;
09. Função objetivo: calcula penalidades e grava solução em arquivos.

Abaixo seguem descritas as etapas da fase 1:

##### **00. Recebe vetor “x” da metaheurística e transforma em vetor permutado:**

Esta é a 1ª das etapas principais do algoritmo. Nela, o algoritmo recebe um vetor “x” da metaheurística, resultado de permutações internas dela.

Precisa-se passar os valores de “x” para o vetor permutado do algoritmo, mas não se trabalha direto com esse vetor “x” original. Trabalha-se com um “novo x”. Isto, porque o vetor “x” é composto de números reais dado que esse é o tipo de retorno das metaheurísticas. Mas, como se precisa de números inteiros, para ultrapassar esse obstáculo é criado um “novo x” onde se ordena as posições do vetor “x” de forma crescente. Dessa forma tem-se uma sequência de posições, variando de 1 a “n + m”.

Adota-se como “novo x” os números das posições ordenadas do “x” original e se passa para o algoritmo esse vetor “ordenado” como sendo o vetor permutado. Com esse “artifício”, sempre ter-se-á um vetor com valores de 1 a  $n\_vetor$ , sendo  $n\_vetor$  = dimensão do vetor.

##### **01. Localiza a Posição das Instalações no Vetor Permutado**

Esta é uma etapa intermediária necessária para a etapa 3 (criar um vetor de rotas). Aqui identificam-se as posições em que se encontram cada uma das instalações, exemplo: vetor “x” com 10 pontos e 3 instalações (nºs. 11, 12 e 13), a instalação nº. 11, está na posição 9 do vetor (é a 2ª instalação a aparecer no vetor), enquanto a instalação nº.12, está na

posição 4 do vetor (é a 1º instalação a aparecer no vetor) e a instalação nº.13, está na posição 12 do vetor (é a 3º instalação a aparecer no vetor), conforme tabela abaixo. Para essa situação tem-se como retorno o vetor c (9, 4, 12).

**Tabela 3 - Representação Esquemática do Vetor "x"**

PONTOS										Instalações		
1	2	3	4	5	6	7	8	9	10	11	12	13
3	2	4	12	10	7	9	8	11	5	6	13	1

## 02. Identifica Ordenação das Instalações no Vetor Permutado

Esta é outra etapa intermediária necessária para a etapa 3 (criar um vetor de rotas). Aqui identifica-se ordenação das instalações no vetor e tem-se como retorno o vetor c (2, 1, 3), sendo que a instalação nº. 11, é a 2ª instalação a aparecer no vetor, enquanto a nº.12, é a 1ª instalação a aparecer no vetor, já a nº.13, é a 3ª instalação a aparecer no vetor.

## 03. Cria um vetor de rotas

Esta é a 2ª das etapas principais do algoritmo. Nela, transforma-se o vetor permutado em uma sequência de rotas sendo que o vetor deve iniciar com o número de uma instalação e ter uma sequência do tipo: rota 1, rota 2, ..., rota n.

Esta função puxa para a 1ª posição a 1ª instalação que aparece no vetor. Dessa forma o vetor passa a iniciar com uma instalação. Feito isso, ele puxa para as posições seguintes toda a sequência após a 1ª instalação e depois disso, a função transfere para o final, os primeiros elementos do vetor que estavam nas posições do vetor anteriores à primeira instalação. Assim os pontos do início do vetor vão para o final. No exemplo, tem-se:

Vetor x original →	<b>x = c (3, 2, 4, 12, 10, 7, 9, 8, 11, 5, 6, 13, 1)</b>
Vetor x de rotas →	<b>x = c (12, 10, 7, 9, 8, 11, 5, 6, 13, 1, 3, 2, 4)</b>

**Figura 1 - Distribuição dos vetores originais e vetor de rotas após execução da etapa 02.**

## 04. Identifica posição de início de cada rota no vetor de rotas e sua ordem

Esta etapa intermediária é similar à etapa 1 e é necessária para a etapa 06 (Criar Rotas na forma de Gotas). Em sua execução identificam-se as posições em de cada uma das instalações no vetor de rotas. No exemplo, tem-se:

Vetor de Rotas →	<b>x = c (12, 10, 7, 9, 8, 11, 5, 6, 13, 1, 3, 2, 4)</b>
------------------	--

**Figura 2 - Exemplo do vetor de rotas no momento da execução do passo 04**

Nessa etapa, tem-se como retorno o vetor c (6, 1, 9), que possui as seguintes características: instalação nº. 11, está na posição 6 do vetor (é a 2º instalação a aparecer no vetor), a nº. 12, está na posição 1 do vetor (é a 1ª instalação a aparecer no vetor) e a nº. 13, está na posição 9 do vetor (é a 3º instalação a aparecer no vetor).

Em seguida, identifica-se a ordenação das instalações no vetor de rotas, que é uma sub-etapa, similar à etapa 2. Portanto, tem-se como retorno o vetor c (2, 1, 3) com a seguinte configuração: instalação nº. 11, é a 2ª instalação a aparecer no vetor de rotas, enquanto a nº. 12, é a 1ª instalação e a nº. 13, é a 3ª a aparecer no vetor de rotas.

## 05. Identifica nº. de pontos em cada rota



Esta é uma etapa intermediária necessária para a etapa 6 (criar rotas na forma de gotas). Ela faz a contagem de n°. de pontos por rota e identifica a maior n°. de pontos em uma rota. No exemplo, tem-se como retorno o vetor  $c$  (5, 3, 5), sendo que a 1ª rota do vetor de rotas (instalação n°. 12) tem 5 pontos, a 2ª rota (instalação n°. 11) tem 3 pontos e a 3ª rota (instalação n°. 13) tem 5 pontos, conforme figura 3.

Em seguida, identifica-se o tamanho máximo dessas rotas onde tem-se como retorno o valor 5.

Vetor de Rotas →  $x = c$  (12, 10, 7, 9, 8, 11, 5, 6, 13, 1, 3, 2, 4)

Figura 3 - Vetor de rotas durante a execução do passo 05

### 06. Cria rotas em "gotas" (rotas circulares)

Esta é a 3ª das etapas principais do algoritmo. Nessa etapa, todas as rotas devem iniciar e terminar no seu ponto de origem (formar um circuito) e para que isso ocorra, identifica-se a posição de início de cada rota no vetor de rotas. No exemplo, tem-se:

Vetor de Rotas →  $x = c$  (12, 10, 7, 9, 8, 11, 5, 6, 13, 1, 3, 2, 4)

Figura 4 - Vetor de rotas durante a execução do passo 06

Tem-se como retorno desse passo o vetor  $c$  (1, 6, 9), que possui a seguinte configuração: a 1ª rota do vetor de rotas (instalação n°. 12) inicia na posição 1, a 2ª rota (instalação n°. 11) inicia na posição 6, a 3ª rota (instalação n°. 13) inicia na posição 9.

Em seguida, acrescenta-se o ponto de origem de cada rota ao final da rota e cada uma delas passa a formar um circuito. Diz-se que fica no formato de uma "gota", sendo que as rotas contidas no vetor tem formato de gotas, conforme figura 5.

Vetor de Rotas →  $x = c$  (12, 10, 7, 9, 8, 11, 5, 6, 13, 1, 3, 2, 4)  
 Vetor com Rotas em Gotas →  $x = c$  (12, 10, 7, 9, 8, 12, 11, 5, 6, 11, 13, 1, 3, 2, 4, 13)

Figura 5 - Resultado da execução do passo 06

### 07. Identifica início/fim de cada rota no vetor de rotas em gotas

Esta é uma etapa intermediária necessária para a etapa 8 (cálculo de distâncias), onde identificam-se as posições limites das rotas no vetor com rotas em gotas. No exemplo, tem-se o resultado demonstrado na figura 6, que possui a seguinte configuração: a rota da instalação n°. 11, está entre as posições 7 e 10, e possui 3 pontos; a rota da n°. 12, está entre as posições 1 e 6, e possui 5 pontos; e por fim, a rota da instalação n°. 13, está entre as posições 11 e 16, e possui 5 pontos.

Vetor com Rotas em Gotas →  $x = c$  (12, 10, 7, 9, 8, 12, 11, 5, 6, 11, 13, 1, 3, 2, 4, 13)

Tem-se como retorno a Matriz abaixo:

11	7	10	3
12	1	6	5
13	11	16	5

Figura 6 - Resultado da execução do passo 07

### 08. Função objetivo: calcula distâncias

Nessa etapa, cria-se uma função preliminar: cálculo de distância euclidiana entre dois pontos, que é representada pela função  $Dist()$ . Nela são feitos os cálculos de distâncias

para cada trecho de rota. Deste bloco são retornadas duas saídas que estão representadas na figura 7:

- Vetores de Distâncias por Trecho por Rota	[Distances_Vetor]
- Vetor de Distâncias Totais por Rota	[Distance_Rota]

**Figura 7 - Vetores retornados na etapa de cálculo de distâncias**

### 09. Função objetivo: calcula penalidades e grava solução em arquivos

Esta é a 5ª e última das etapas principais da fase 1 do algoritmo. Ela inclui o cálculo de penalidades por excesso de demanda, não atendimento, acessibilidade e segurança na rota entre os clientes e a instalação somando os impactos de cada trecho de rota, sendo retornadas cinco saídas deste bloco, conforme a figura 8:

- Demanda excedente global da Operação	[Demanda_excedente]
- Demanda não atendida global da Operação	[Demanda_nao_atendida]
- Penalidade por segurança global da Operação	[Penalidade_seguranca]
- Penalidade por acessibilidade Global da Operação	[Penalidade_acessibilidade]
- Distância Global da Operação em Estudo	[Distance_Global]

**Figura 8 - Vetores retornados na etapa de cálculo de penalidades**

Em seguida grava resultados obtidos na fase 1 em arquivos, listados na figura 9. Na linguagem R, utiliza-se comando *write.table()*, com parâmetro *append* para que os resultados obtidos a cada interação sejam incluídos sem que os registros já gravados nos arquivos sejam excluídos.

- [Distances_Vetor]	→ arquivo "Distancias_por_Trecho_de_Rota2.csv"
- [Distance_Rota]	→ arquivo "Distancias_por_Rota2.csv"
- [Distance_Global]	→ arquivo "Distancia_Global2.csv"
- [Rotas_e_Dist]	→ arquivo "Rotas_Circulares2.csv"
- [Limites_Rota]	→ arquivo "Pontos_por_Rota2.csv"
- [clientes_instalacoes]	→ arquivo "matriz_instalacoes.csv"
OBS: [Rotas_e_Dist] = Vetor com Rotas em Gotas	
[Limites_Rota] = Matriz com Limites das Rotas e No. de Pontos por Rota	
[clientes_instalacoes] = Vetor com os dados de atendimento dos clientes pelos fornecedores	

**Figura 9 - Arquivos retornados na etapa 09, ao final da Fase 1**

### 4.3. FASE INTERMEDIÁRIA: Bloco único de transição entre fases

A Fase Intermediária basicamente, recupera e adapta os resultados da fase 1 para que possam ser utilizados como insumos para a fase 2.

A descrição dos passos desta fase intermediária é apresentada na sequência.

00. Extrai dados da fase 1, capturando os dados dos arquivos de rotas circulares e os dados de distâncias por trecho de rota;
01. Cria um repositório dessas informações já tratadas, transformando-as em bases de dados consolidadas com variáveis calculadas necessárias para a fase 2
02. Cria também, bases de apoio enriquecidas com as informações dos clientes atendidos nas rotas circulares e os dados de coordenadas, de demanda e de penalidades que serão utilizadas nos algoritmos.

### 4.4. FASE 2: Algoritmo de definição de Fornecedores por Instalação

Nesta fase, ocorre a definição dos fornecedores de produtos que deverão atender cada uma das instalações logísticas já definidas na fase 1.

Os passos desta fase são apresentados abaixo:

00. Recebe vetor “x” da metaheurística (solução) e transforma em vetor permutado
01. Cria matriz de combinações de instalações x fornecedores
02. Cria matriz de solução para as combinações definidas pela metaheurística
03. Calcula os parâmetros da função objetivo que sensibilizarão o custo global e Grava Solução em Arquivos

Abaixo seguem descritas as etapas da fase 2:

### 00. Recebe vetor “x” da metaheurística e transforma em vetor permutado

Esta é a 1ª das etapas principais do algoritmo da fase 2, sendo que ela recebe um vetor “x” da metaheurística, resultado de permutações internas da metaheurística e adicionalmente precisa-se passar os valores de “x” para o vetor permutado do algoritmo. Apesar disso, não se trabalha direto com esse vetor “x” original, mas sim, com um “novo x”. Isto, porque o vetor “x” é composto de números reais, assim como visto no passo 00 da fase 1, as metaheurísticas sempre devolvem números desse tipo, mas são necessários números inteiros para utilizar o código. Assim como é feito na fase 1, para ultrapassar esse obstáculo, cria-se um “novo x” ordenando as posições do “x” em ordem crescente de valores. Dessa forma tem-se uma sequência de posições, variando de 1 a “n + m”, onde adota-se como “novo x” os números das posições ordenadas do “x” original.

É passado para o algoritmo esse vetor "ordenado" como sendo o vetor permutado com esse “artifício”, sempre ter-se-á um vetor com valores de 1 a n\_vetor (n\_vetor = dimensão do vetor), dessa vez, o procedimento é realizado para escolha dos fornecedores potenciais para atendimento das instalações.

### 01. Cria matriz de combinações de instalações x fornecedores

Esse passo define valores para as variáveis de custos das penalidades e frete, enriquece matriz com todas as possibilidades de atendimento entre instalações e fornecedores que será utilizada como apoio para a matriz de solução proposta pelo ABC, calcula o frete, a demanda atendida e a capacidade para cada uma das combinações. Exemplo: matriz dados com 2 fornecedores (n<sup>os</sup>. 1 e 2) e 3 instalações (n<sup>os</sup>. 11, 12 e 13), conforme tabela abaixo:

**Tabela 4 - Representação Esquemática da matriz de combinações**

instalacao	fornecedor	distancia	Frete (distancia * custo)	clientes_inst	demanda	capacidade	pen_seguranca (seg. inst. * seg. forn.)	pen_acessibilidade (Aces. inst. * Aces. forn.)
11	1	100	1000	10	100	110	5*5=25	1*3=3
12	1	200	2000	5	200	110	1*5=5	8*3=24
13	1	150	1500	8	300	110	6*5=30	3*3=9
11	2	320	3200	10	100	350	5*7=35	1*9=9
12	2	41	410	5	200	350	1*7=7	8*9=72
13	2	50	500	8	300	350	6*7=42	6*9=54

No exemplo acima, a instalação n<sup>o</sup>. 11, aparece nas linhas 1 e 4 com uma linha para cada fornecedor e com seus parâmetros calculados para os fornecedores 1 e 2.

cliente	fornecedor	distancia	frete	demanda	capacidade
1 1	1	40.607881	406.07881	0	405
2 1	2	46.010868	460.10868	0	438
3 1	3	47.010637	470.10637	0	228
4 1	4	21.213203	212.13203	0	452
5 1	5	44.721360	447.21360	0	353

**Figura 10 - Recorte de tela do resultado parcial da Matriz de dados (Combinações)**

## 02. Cria matriz de solução para as combinações definidas pela metaheurística

Esse passo recebe os fornecedores e as instalações escolhidos pela função ABC e captura a combinação calculada anteriormente na matriz de apoio fornecedores que será utilizada como apoio para a matriz de solução proposta pelo ABC e calcula a demanda atendida e a não atendida para a combinação escolhida e salva na matriz solução.

```

if ((vetor_permutado[i] > 0) && (vetor_permutado[i] <= qtd_fornecedor)){
  retorno = determina_fornecedor(i, vetor_permutado[i])
  matriz_solucao = rbind(matriz_solucao, retorno)
  forns <- which(matriz_dados[,2] == retorno[,2])
  cli <- which(matriz_dados[,1] == retorno[,1])
  for (j in forns){
    matriz_dados[j,6] = retorno[,7]
  }
  matriz_forn[which(matriz_forn[,1] == retorno[,2]),2] = retorno[,7]
  for (j in cli){
    matriz_dados[j,5] = retorno[,9]
  }
  matriz_cli[which(matriz_cli[,1] == retorno[,1]),2] = retorno[,9]
} else{
  matriz_solucao = rbind(matriz_solucao, c(i,0,0,0,matriz_cli[i,2]),0,0,0,matriz_cli[i,2])
}
}
matriz_solucao <- matriz_solucao[-1,]

```

Figura 11 – Recorte de tela da chamada da função de cria a matriz de solução

```

#cria Polina
determina_fornecedor <- function(i, j){
  endereco = which((matriz_dados[,1] == i) & (matriz_dados[,2] == j))
  temp <- matrix(ncol = 9, nrow = 1)
  temp[,1] = i
  temp[,2] = matriz_dados[endereco,2]
  temp[,3] = matriz_dados[endereco,3]
  temp[,4] = matriz_dados[endereco,4]
  temp[,5] = matriz_dados[endereco,5]
  temp[,6] = matriz_dados[endereco,6]

  if (matriz_dados[endereco,5] > matriz_dados[endereco,6])
  {temp[,7] = 0}
  else{temp[,7] = matriz_dados[endereco,6] - matriz_dados[endereco,5]}
  if (matriz_dados[endereco,6] > matriz_dados[endereco,5])
  {temp[,8] = matriz_dados[endereco,5]}
  else{temp[,8] = matriz_dados[endereco,6]}
  if (matriz_dados[endereco,5] > matriz_dados[endereco,6])
  {temp[,9] = matriz_dados[endereco,5] - matriz_dados[endereco,6]}
  else{temp[,9] = 0}
  return(temp) }

```

Figura 12 – Recorte de tela da função que cria a matriz de solução

Como pode-se ver no exemplo abaixo, tabela 5: em uma matriz solução com demandas calculadas, supondo que a metaheurística tenha definido o fornecedor n°. 1 para as instalações n°. 11 e 12 e o fornecedor n°. 2 para a instalação n°. 13, pode-se observar que o cálculo de demanda atendida corresponde ao valor da *capacidade\_inst* se a demanda for maior que ela, caso contrário será igual a demanda se a capacidade\_inst for maior que a demanda. Já no caso de a demanda ser inferior à *capacidade\_inst*, a demanda excesso é igual a *capacidade\_inst* – demanda. A coluna capacidade restante da instalação recebe o valor da ocorrência anterior da mesma instalação, como ocorre nas linhas 1 e 2, em que a primeira capacidade é de 110 com a restante sendo 10 após atender uma demanda de 100. a linha 2 recebe a capacidade de 10, que é a nova capacidade após o primeiro atendimento.

Tabela 5 - Representação Esquemática da matriz de soluções

instalacao	fornecedor	distancia	frete (distancia * custo)	clientes_inst	demanda	capacidade_inst	capacidade restante	demanda atendida	demanda excesso	seq_inst * 200 + forn	seq_inst * 200 + forn - 1
11	1	100	1000	10	100	110	10	110	0	5*5+25	1*3+3
12	1	200	2000	5	200	10	0	10	190	1*5+5	8*3+24
13	2	50	500	8	300	350	50	300	0	6*3+42	6*5+54

cliente	fornecedor	distancia	frete	demanda	capacidade	capacidade restante	demanda atendida	demanda excesso
1	1	3	4701054	4701054	10	238	228	10
2	2	16	6661081	6661081	7	367	360	7
3	3	15	4841487	4841487	13	286	273	13
4	4	12	2529822	2529822	19	132	113	19
5	5	19	5336666	5336666	26	95	69	26
6	6	18	3080584	3080584	3	403	400	3
7	7	10	5122499	5122499	5	195	190	5
8	8	12	3701351	3701351	9	113	104	9
9	9	20	4310452	4310452	16	280	244	16
10	10	18	5651548	5651548	16	400	384	16
11	11	6	6514599	6514599	60	448	389	60
12	12	9	3176476	3176476	60	313	253	60
13	13	1	4609772	4609772	60	465	405	60

Figura 13 – Recorte de tela da matriz Solução

### 03. Calcula os parâmetros da função objetivo que sensibilizarão o custo global e grava solução em arquivos

Esta é a 3ª das etapas principais do algoritmo. Nela, os parâmetros são calculados de forma consolidada e tem-se como retorno oito saídas deste bloco, que estão representados na figura 14, abaixo:

- Custo de frete global da Operação	[frete_global]
- Total de clientes atendidos pela Operação	[clientes_global]
- Demanda global da Operação	[demanda_global]
- Capacidade global dos fornecedores da Operação	[capacidade_global]
- Demanda excedente global da Operação	[Demanda_excedente]
- Demanda não atendida global da Operação	[Demanda_nao_atendida]
- Demanda atendida global da Operação	[demanda_atendida]
- Custo Global da Operação em Estudo	[Custo_Global]

Figura 14 - Retorno do bloco 04

Vale destacar os seguintes conceitos sobre os dados retornados:

- Frete: somatória da coluna frete da matriz solução;
- Clientes: somatória da coluna clientes\_inst da matriz solução;
- Demanda: somatória da coluna demanda da matriz solução;
- Demanda atendida: somatória da coluna demanda atendida da matriz solução;
- Demanda excedida: somatória da coluna demanda excesso para os casos em que coluna demanda atendida é maior que zero na matriz solução;
- Demanda não atendida: somatória da coluna demanda excesso para os casos em que a coluna demanda atendida é igual a zero na matriz solução;
- Segurança: somatória do cálculo de penalidade de segurança multiplicado pelo custo da penalidade;
- Acessibilidade: somatória do cálculo de penalidade de acessibilidade multiplicado pelo custo da penalidade;
- Custo global: somatória dos parâmetros calculados anteriormente para o público do estudo;

Em seguida há a gravação dos resultados em arquivo, sendo que em linguagem R, utiliza-se comando *write.table()*, com parâmetro *append* para não substituir os resultados gravados nas interações anteriores. Os resultados gravados são os 4 destacados na figura 15, abaixo:

- [distancia_Global]	→ arquivo "Distancia_Global_forn.csv"
- [Custo_Global]	→ arquivo "Custo_Global_forn.csv"
- [matriz_dados]	→ arquivo "combinacoes_custos.csv"
- [matriz_solucão]	→ arquivo "matriz_combinacoes.csv"

Figura 15 - Arquivos gerados pela fase 2

#### 4.5. Penalidades por falta ou excesso, acessibilidade e segurança

No problema de localização de instalações pode-se ter quatro tipos de penalidades: penalidade por falta de atendimento da demanda total de um ponto de atendimento; penalidade por exceder a oferta total de carga de um ponto de abastecimento; penalidade pelo nível de problemas de acessibilidade nas rotas até os pontos; e penalidade pelo nível de problemas de segurança nas rotas até os pontos.

É considerado que se pode ter o mesmo tipo de penalidade, seja para demanda não atendida, seja para oferta excedida. Visto que a penalidade seria equivalente a um custo adicional de transporte que seria necessário para atender a demanda não atendida ou para suprir a oferta excedida de um fornecedor (ponto de abastecimento).

Tem-se um custo adicional de transporte para outro fornecedor atender essa demanda não atendida. Vejamos abaixo:

- Penalidade por falta de atendimento da demanda = custo fixo/unid. x n°. de unidades não atendidas;

Tem-se um Custo Adicional de Transporte para Outro Fornecedor suprir essa Oferta que foi excedida:

- Penalidade por exceder oferta do fornecedor = custo unitário de excesso x n°. de unidades excedidas;

O custo adicional de transporte (CATunid) por unidade de demanda (ton, por exemplo) será a penalidade, seja para demanda não atendida, seja para oferta excedida, conforme imagem abaixo:

$$\begin{aligned}
 CAT_{UNID} &= \text{Custo Adicional de Transporte} / \text{Unidade de Demanda} \\
 CAT_{UNID} &= \text{Custo Total de Transporte} / \text{Demanda Total} \\
 \text{Custo Unitário de Excesso ou de Falta} &= CAT_{UNID}
 \end{aligned}$$

**Figura 16 - Representação das penalidades por demanda**

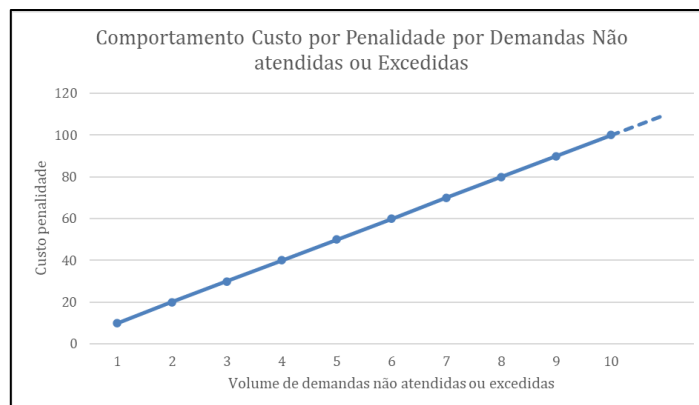
As penalidades correspondentes a demanda possuem o comportamento de uma reta de função de primeiro grau. Isso ocorre, pois o prejuízo gerado por demandas não atendidas e excedentes, na prática, tende a crescer conforme seu volume vai aumentando, evoluindo conforme ele vai se acumulando.

Abaixo segue a representação da reta da função expressa pela seguinte fórmula:

$$y = \text{Custo da penalidade} * \text{Volume de demanda não atendida ou excedida}$$

$$\text{Custo da penalidade} = 10$$

$$y = 10 * x$$



**Figura 17 - Gráfico de comportamento da reta de penalidade por demandas não atendidas**

Também é considerado que se pode ter o mesmo tipo de penalidade, seja para segurança quanto para acessibilidade.

A penalidade seria equivalente a um custo adicional de transporte que seria necessário para suprir a necessidade de investimento em segurança e os custos gerados por conta dos problemas com acessibilidade considerando os referenciais dos pontos de partida e de destino.

Nesse contexto, a escala de problemas de acessibilidade e segurança variam de 1 a 10, sendo 1 considerado como ponto sem problemas e 10 um ponto com o maior nível de problemas possível.

Tem-se um custo adicional de transporte para que uma instalação/fornecedor atenda determinado cliente/instalação dependendo do nível de problemas identificados em ambos os pontos (origem e destino):

- Penalidade por nível de problemas de segurança = custo fixo/unid.  $\times \log_{10}$  (nível de problemas de segurança no ponto a  $\times$  nível de problemas de segurança no ponto b);

Tem-se um custo adicional de transporte para que uma instalação/fornecedor atenda determinado cliente/instalação dependendo do nível de problemas identificados em ambos os pontos (origem e destino):

- Penalidade por nível de problemas de acessibilidade = custo fixo/unid.  $\times \log_{10}$  (nível de problemas de acessibilidade no ponto a  $\times$  nível de problemas de acessibilidade no ponto b);

Pode-se observar que esses dois tipos de penalidade envolvem dois pontos, o de partida, que pode ser a instalação ou o fornecedor e o ponto de destino, que pode ser o cliente ou a instalação, a depender da etapa logística. Para construir a fórmula, considera-se que os valores dos pontos devem ser multiplicados, pois está-se atribuindo os níveis de problemas referenciais dos pontos a um gradiente de redução ou aumento de problemas ao longo da rota entre os dois pontos.

Outro item a se considerar é que diferentemente do comportamento de reta de função de primeiro grau observado na penalidade por demandas não atendidas ou excedidas, as penalidades por nível de problemas de segurança e acessibilidade se comportam como uma curva de função logarítmica. Isso ocorre porque os gastos com investimentos com os problemas citados, diferentemente do comportamento de prejuízo gerados pelas demandas não atendidas, não tendem ao infinito. Esses gastos tendem a subir conforme seja mais problemático o atendimento a determinada região, mas estão limitados à capacidade de investimento e ofertas de serviços de segurança, que tendem a reduzir conforme o custo aumenta.

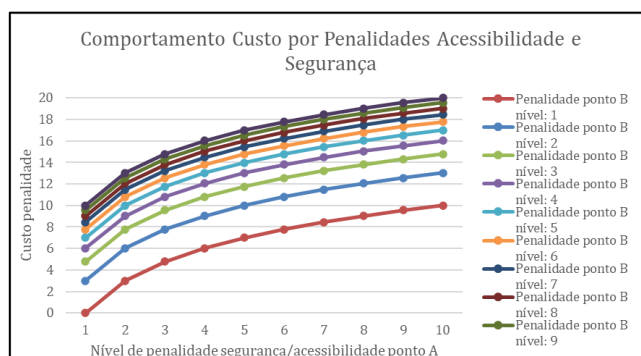
Portanto, a função dos dois itens (acessibilidade e segurança) é representada pela seguinte fórmula:

$$y = \text{Custo da penalidade} * \log_{*} \left( \frac{\text{Nível de problemas ponto A}}{\text{Nível de problemas ponto B}} \right)$$

$$\text{Custo da penalidade} = 10$$

$$y = 10 * \log_{*} \left( \frac{\text{Nível de problemas ponto A}}{\text{Nível de problemas ponto B}} \right)$$

Dada a escala de 1 a 10, o comportamento das retas para essas penalidades e representado pelo gráfico abaixo:

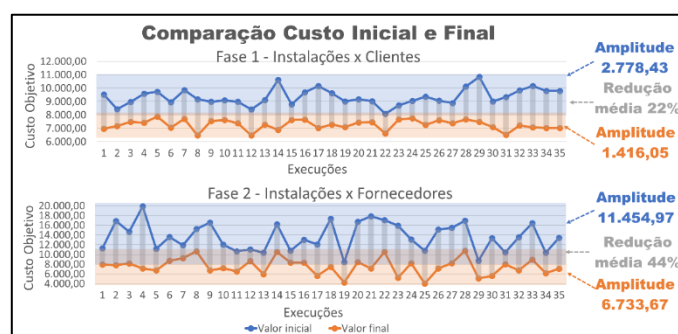


**Figura 18 - Gráfico de comportamento da reta de penalidade por problemas de acessibilidade e segurança**

## 5. Conclusões e Recomendações

Realizando 35 execuções do algoritmo para se ter uma comparação de um volume relevante de testes também foi possível observar que a amplitude de variação entre os resultados iniciais e finais sempre é reduzida, independente do custo da primeira solução proposta pela *Artificial Bee Colony*, que é utilizado nas duas etapas.

Considerando que a ABC inicia sua execução de um ponto aleatório, podendo ser um ponto favoravelmente mais baixo ou desfavoravelmente mais alto, a amplitude média inicial na fase 1 foi de 2.778,43 u.m., e na fase 2 foi de 11.454,97 u.m., mas em todos os casos, sem exceção, o custo final apresentou redução em relação ao inicial, sendo que a fase 1 apresentou uma média de 1.416,05 u.m. e a fase 2 teve uma média de 6.733,67 u.m., o resultado observado foi: redução média dos custos da fase 1 em 22%, enquanto na fase 2 houve redução média de 44% em relação aos valores iniciais.



**Figura 19 - Gráficos de comparação dos custos nas execuções da fase 1 e fase 2**

Logo, pode-se chegar nessa conclusão, de que com o uso da metaheurística, neste caso em particular a *Artificial Bee Colony*, apesar de não ser garantida a melhor solução definitiva, dada a dependência da escolha aleatória do ponto inicial, pode-se ver que em todos os casos ele reduz o custo inicial. O que a coloca dentro do espectro de soluções possíveis para o problema de distribuição das instalações.

## Referências

BARRETO, S., FERREIRA, C., PAIXÃO, J., SANTOS, B. (2007). **Using clustering analysis in a capacitated location-routing problem**, European Journal of Operational Research, Vol. 179, Issue 3, 2007, p. 968-977, ISSN 0377-2217. Disponível em <<https://doi.org/10.1016/j.ejor.2005.06.074>> Acesso em 15/11/2020.



- BOUSSAÏD, I., LEPAGNOT, J., SIARRY, P. (2013). **A survey on optimization metaheuristics** Information Sciences, Vol. 237, 2013, p. 82-117, ISSN 0020-0255, Disponível em <<https://doi.org/10.1016/j.ins.2013.02.041>> Acesso em 05/06/2021.
- BRANDEAU, M. L., CHIU, S. S. (1989). **An Overview of Representative Problems in Location Research.** *Management Science*, Vol. 35, no. 6, 1989, p. 645–674. *JSTOR*, Disponível em <[www.jstor.org/stable/2632113](http://www.jstor.org/stable/2632113)> Acesso em 12/04/2021.
- CHRISTOFIDES, N.; MINGOZZI, A. and TOTH, P. (1979). **The Vehicle Routing Problem. In: Combinatorial Optimization**, pp. 315-338, N. Christofides, A. Mingozzi, P. Toth and C. Sandi (eds.), Wiley, Chichester.
- CORREIO BRAZILIENSE (2020). **Atrasos em compras on-line se tornam mais comuns; saiba como agir.** Disponível em: <[https://www.correiobraziliense.com.br/app/noticia/cidades/2020/05/25/interna\\_cidade/857956/atrasos-em-compras-on-line-se-tornam-mais-comuns-saiba-como-agir.shtml](https://www.correiobraziliense.com.br/app/noticia/cidades/2020/05/25/interna_cidade/857956/atrasos-em-compras-on-line-se-tornam-mais-comuns-saiba-como-agir.shtml)>. Acesso em: 02/11/2020.
- CRAN (2021). The Comprehensive R Archive Network. R-Project, Disponível em: <https://cran.r-project.org>
- DASKIN, M.S. (2008). **What you should know about location modeling.** *Naval Research Logistics*, Vol. 55, 2008, p. 283-294. Disponível em <<https://doi.org/10.1002/nav.20284>> Acesso em 15/11/2020.
- KARABOGA, D. (2005). **An Idea Based on Honey Bee Swarm for Numerical Optimization**, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri/Türkiy, 2005. Disponível em <<http://lia.disi.unibo.it/Courses/SistInt/articoli/bee-colony1.pdf>> Acesso em 12/04/2021.
- KARABOGA, D., AKAY, B. (2009). **A comparative study of Artificial Bee Colony algorithm**, *Applied Mathematics and Computation*, Vol. 214, Issue 1, 2009, p. 108-132, ISSN 0096-3003, Disponível em <<https://doi.org/10.1016/j.amc.2009.03.090>> Acesso em 05/03/2021.
- MASUTTI, T. A. S. (2016). **Um framework inspirado no comportamento coletivo de abelhas para a solução de problemas de roteamento de instalações.** 2016. 119 f. Dissertação (Engenharia Elétrica) - Universidade Presbiteriana Mackenzie, São Paulo. Disponível em <<http://tede.mackenzie.br/jspui/bitstream/tede/3123/5/THIAGO%20AUGUSTO%20SOARES%20MASUTTI.pdf>> Acesso em 20/04/2021.
- ONU NEWS (2020). **Em Dia Mundial da Estatística, ONU defende que pandemia aumentou relevância de dados.** Disponível em: <<https://news.un.org/pt/story/2020/10/1730132>> Acesso em: 26/10/2020.
- 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*, Vol. 184, Issue 3, 2008, p. 817-848, ISSN 0377-2217, Disponível em <<https://doi.org/10.1016/j.ejor.2006.12.044>> Acesso em 15/11/2020.
- Riza, L. S. ; Iip; Nugroho, E. P.; Prabowo, M. B. A.; Junaeti, E.; Abdullah, A. G. (2019). **Package ‘metaheuristicOpt’.** CRAN - The Comprehensive R Archive Network. R-

- Project, Disponível em: <https://cran.r-project.org/web/packages/caret/index.html>. Acesso em 05/02/2020.
- RStudio (2021). RStudio the premier IDE for R. Disponível em: <https://www.rstudio.com/products/rstudio/>
- SAHIN, O., AKAY, B., KARABOGA, D. (2021). **Archive-based multi-criteria Artificial Bee Colony algorithm for whole test suite generation**, Engineering Science and Technology, an International Journal, Vol. 24, Issue 3, 2021, p. 806-817, ISSN 2215-0986, Disponível em <<https://doi.org/10.1016/j.jestch.2020.12.011>> Acesso em 06/05/2021.
- SEBRAE (2020). **Coronavírus: o impacto nas vendas online**. Disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/artigos/coronavirus-o-impacto-nas-vendas-online,ed84f8e520f71710VgnVCM1000004c00210aRCRD>> Acesso em: 02/11/2020.
- SILVA, A. (2020). **Centros de distribuição como vantagem competitiva**. Disponível em: <<https://revista.pgsskroton.com/index.php/rcger/article/view/2710>> Acesso em: 02/11/2020.
- VALE, M. N. DO (2006). **Agrupamentos de dados: avaliação de métodos e desenvolvimento de aplicativo para análise de grupos**, publicação acadêmica, Pontifícia Universidade Católica Do Rio De Janeiro - PUC-Rio, 2006. Disponível em <<https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=resultado&nrSeq=7975@1>> Acesso em 15/11/2020.
- VRP-REP (2021). “**Base de Dados Christofides et al. 1979**”. The vehicle routing problem repository. Disponível em: <http://www.vrp-rep.org/datasets/item/2014-0002.html>. Acesso em 04/05/2021.