



# Logística Operacional Neoenergia: Algoritmos para Determinar a Localização Ótima de Bases e Escalas de Trabalho

**Tema:** Operação

**Autores:** Rodrigo Manfredini

**Co-Autores:** Leonardo Bezerra, Adriele Correia Ribeiro, Francisco Pereira de Medeiros Neto, José Ivan de Sá Junior, Ramon Reis Silva, Willian Klein Siewerdt

**Empresa:** Companhia de Eletricidade do Estado da Bahia - Neoenergia Coelba

---

## Resumo

Este trabalho apresenta a metodologia de logística operativa aplicada às distribuidoras do grupo Neoenergia. Foram utilizados algoritmos de machine learning, incluindo K-means clássico, Weighted K-means e o algoritmo de otimização iterated greedy. O objetivo desses algoritmos é determinar a localização ideal das bases dos eletricitas, considerando os serviços realizados, tanto para atendimentos emergenciais quanto para notas comerciais. A aplicação dessas metodologias resultou em uma significativa redução nos tempos de deslocamento. Além dos algoritmos de localização de bases, o artigo também aborda a otimização matemática para definir as escalas de trabalho ideais dos eletricitas.

## 1. Introdução

A distribuição de energia elétrica envolve a operação, expansão e manutenção de subestações e redes de média e baixa tensão, com o objetivo de melhorar continuamente a qualidade do fornecimento a custos acessíveis. Nesse contexto, é essencial que as empresas busquem soluções para aprimorar a eficiência operacional e a qualidade do serviço.

Desse modo, a Neoenergia tem aplicado diversos algoritmos ao longo dos últimos anos para as 5 distribuidoras do grupo (Coelba, Elektro, Pernambuco, Cosern e Brasília) a fim de otimizar a alocação de suas bases operativas e as escalas de trabalho, visando atender às ocorrências e notas comerciais com a maior qualidade e no menor tempo possível. A figura 1 mostra a distribuição espacial das ocorrências emergenciais da Elektro (UTD de Campos do Jordão), bem como as bases de apoio.

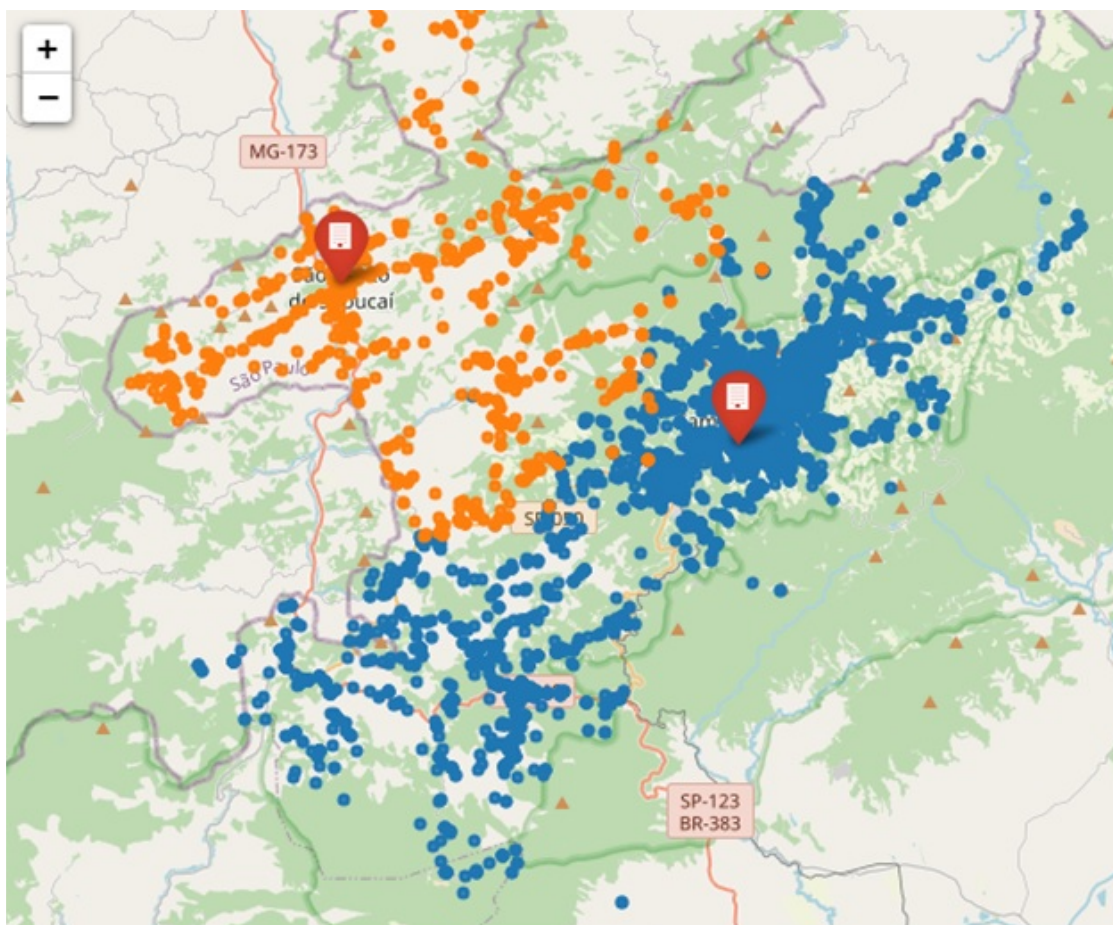


Figura 1: Distribuição das ocorrências emergenciais por base de apoio para a Neoenergia Elektro (UTD de Campos do Jordão). Mapa mostrando as 2 bases de apoio (pinos em vermelho) e as ocorrências em forma de círculos. A cor dos círculos identifica as ocorrências que foram atendidas pela mesma base.

Este trabalho está organizado da seguinte forma: inicialmente, será abordado o tema de aprendizado de máquina, seguido pela apresentação dos algoritmos K-means e K-means ponderado. Em seguida, serão discutidos os modelos de otimização iterated greedy para alocação de bases e GRG (Generalized Reduced Gradient) para a otimização das escalas. Por fim, serão apresentados os resultados e as conclusões.

## 2. Desenvolvimento

### 2.1 Aprendizado de Máquina

O aprendizado de máquina (Machine Learning) é uma subárea da inteligência artificial que utiliza técnicas computacionais para resolver problemas com base em dados e informações históricas, sem exigir modificações significativas no processo principal (Sandhu, 2018). Dessa forma, o aprendizado de máquina envolve o estudo de técnicas poderosas que podem aprender com a experiência. À medida que um algoritmo de aprendizado de máquina acumula mais experiência, geralmente na forma de dados observacionais ou interações com o ambiente, seu desempenho tende a melhorar (Zhang et al., 2021).

A maioria dos autores classifica os algoritmos de aprendizado de máquina em supervisionados ou não supervisionados, embora alguns também incluam a categoria de aprendizado por reforço (reinforcement learning). Este último aprende a partir de dados observando o ambiente, reagindo com ações e recebendo recompensas. A principal diferença entre aprendizado supervisionado e não supervisionado é a presença

de rótulos nos dados de treinamento (Alloghani et al., 2020). No aprendizado supervisionado, o algoritmo recebe um conjunto de exemplos de treinamento com rótulos conhecidos. Já no aprendizado não supervisionado, o algoritmo analisa os dados fornecidos e tenta agrupá-los de alguma forma, formando clusters (Monard & Baranauskas, 2003). Uma diferença importante no aprendizado supervisionado é que, embora muitos algoritmos de agrupamento sejam formulados com algum critério de otimalidade, não há garantia de que a solução ótima global seja alcançada.

Como, para o problema em questão, não atribuímos rótulos aos dados de treinamento, temos um problema de machine learning não supervisionado.

## 2.2 O algoritmo K-means

O algoritmo K-means é uma técnica de aprendizado não supervisionado amplamente utilizada para a análise de agrupamento (clusterização) em ciência de dados e aprendizado de máquina. Seu objetivo é dividir um conjunto de dados em K grupos (ou clusters), de forma que os dados dentro de cada grupo sejam mais semelhantes entre si do que em relação aos dados de outros grupos.

Ele opera em duas etapas principais: atribuição e atualização. Inicialmente, o algoritmo seleciona aleatoriamente K centros, chamados de centroides. Em seguida, ele ajusta esses centroides repetidamente até encontrar a melhor separação entre os dados.

O processo é o seguinte:

1. **Definição do Número de Clusters (K):** O número de clusters K deve ser definido previamente.
2. **Inicialização dos Centroides:** Escolha aleatória dos pontos iniciais que representarão os centros dos clusters.
3. **Atribuição:** Cada ponto de dados é atribuído ao centróide mais próximo, formando os clusters iniciais.
4. **Atualização:** O centróide de cada cluster é recalculado com base na média dos pontos alocados a ele.

Esse processo de atribuição e atualização se repete até que os centroides não mudem mais significativamente, indicando que os clusters estão bem definidos. A ilustração desse processo está na figura 2.

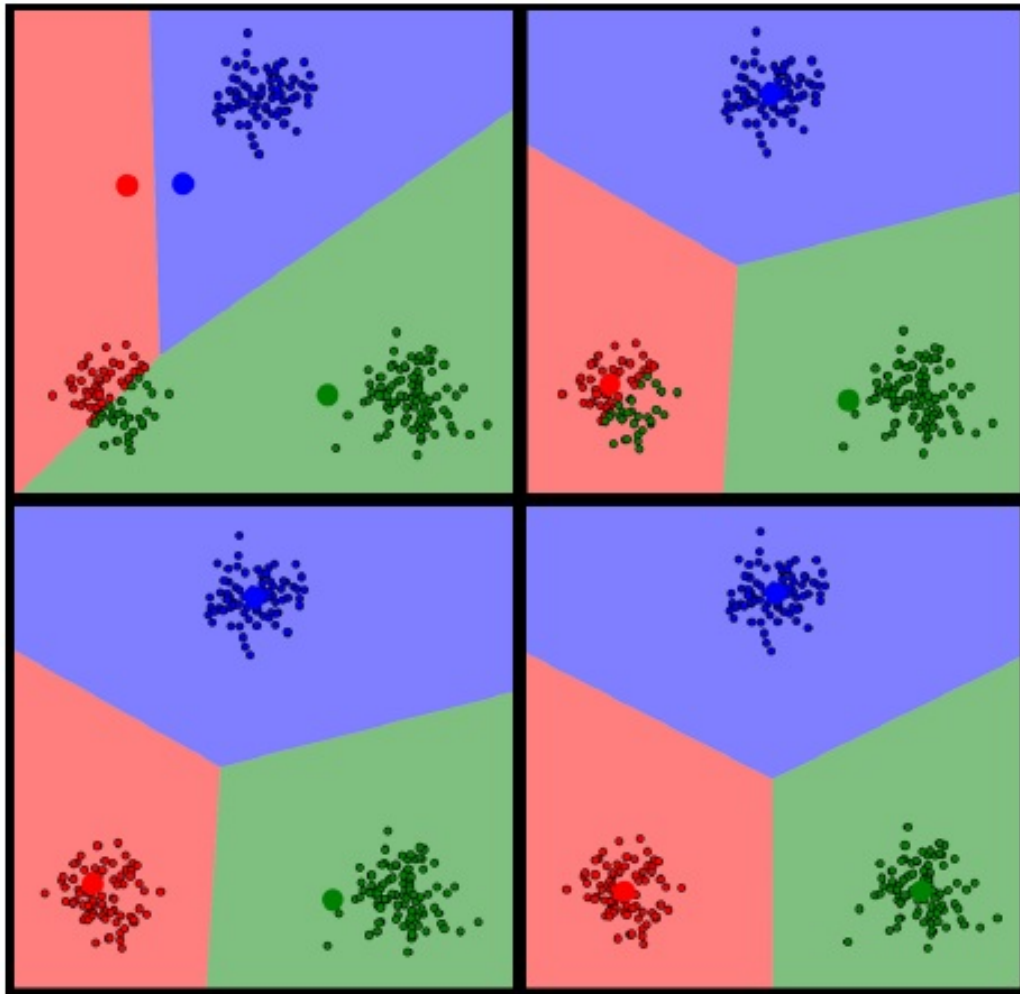


Figura 2: Exemplo de funcionamento do algoritmo K-means. O primeiro quadro representa a inicialização dos centros dos clusters e a atribuição dos pontos a cada cluster. O segundo quadro mostra a atualização dos centróides. O terceiro quadro ilustra a nova atribuição dos pontos após a atualização dos centróides. O processo é repetido até que os centróides parem de se mover ou que seja atingido o número máximo de iterações. O quarto quadro apresenta as posições finais dos centróides (adaptado de <https://www.naftaliharris.com/>).

O conjunto de dados utilizado pelo algoritmo inclui o histórico de ocorrências e/ou notas comerciais, juntamente com suas respectivas latitudes e longitudes. Além disso, são coletadas as localizações das bases atuais para comparar as bases propostas pelo algoritmo com as já existentes. As bases são alocadas exatamente nos centróides de cada um dos clusters formados.

Em alguns casos, ao decidir manter a localização de determinadas bases, o algoritmo é executado utilizando as latitudes e longitudes dessas bases como centróides iniciais. Portanto, o item 2 (Inicialização dos Centróides) do processo, às vezes, não é realizado de maneira aleatória.

Para rodar o algoritmo Kmeans no python, utilizou-se a biblioteca sklearn.

### 2.3 O algoritmo K-means ponderado (Weighted K-means)

O K-Means ponderado é uma variação do K-Means clássico que leva em consideração pesos associados a cada ponto de dados. Esses pesos podem representar a importância ou a frequência dos pontos de dados, sendo útil em situações onde alguns pontos são mais relevantes que outros.



Os passos do K-Means ponderado são semelhantes aos do K-Means clássico, com a diferença principal na etapa de atualização dos centróides. Ao invés de calcular a média simples, utiliza-se a média ponderada no item 4 do algoritmo (Atualização).

Uma das versões comumente utilizadas pela Neoenergia envolve rodar esse algoritmo utilizando como pesos o CHI da ocorrência (cliente hora interrompido) ou a quantidade de clientes interrompidos da ocorrência (ver figura 3). Dessa forma, são atribuídos pesos maiores às ocorrências mais complexas, e as bases são alocadas mais próximas às ocorrências com maior CHI ou horas interrompidas.

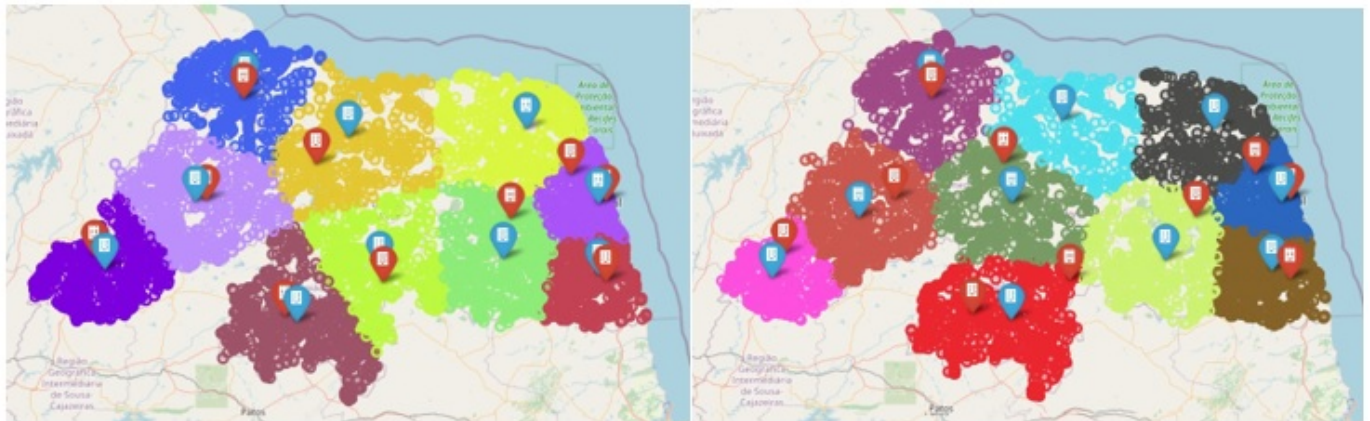


Figura 3: Exemplo do funcionamento dos algoritmos K-Means clássico (a esquerda) e ponderado (a direita) para a Neoenergia Cosern. O algoritmo ponderado sugere locais diferentes para as bases ao considerar o CHI das ocorrências.

## 2.4 Avaliação do Número de Clusters

Uma das limitações do algoritmo K-means é que o número de clusters utilizado não é conhecido a priori e o resultado também pode ser afetado pelo método de inicialização dos centróides. Nesse contexto, o teste do cotovelo é uma técnica bastante utilizada na análise de cluster para determinar o número ideal de clusters ( $k$ ) em um conjunto de dados. São realizados vários agrupamentos para um intervalo de valores de  $k$  (por exemplo, de 1 a 10). Para cada valor de  $k$ , calcula-se a soma dos erros quadrados (SQE). Em seguida, gera-se um gráfico de linha com SQE no eixo y e  $k$  no eixo x. Esse gráfico geralmente assume a forma de um braço, onde o “cotovelo” do braço indica o ponto em que a redução na SQE começa a diminuir significativamente. Esse ponto é considerado o melhor valor de  $k$  (ver figura 4).

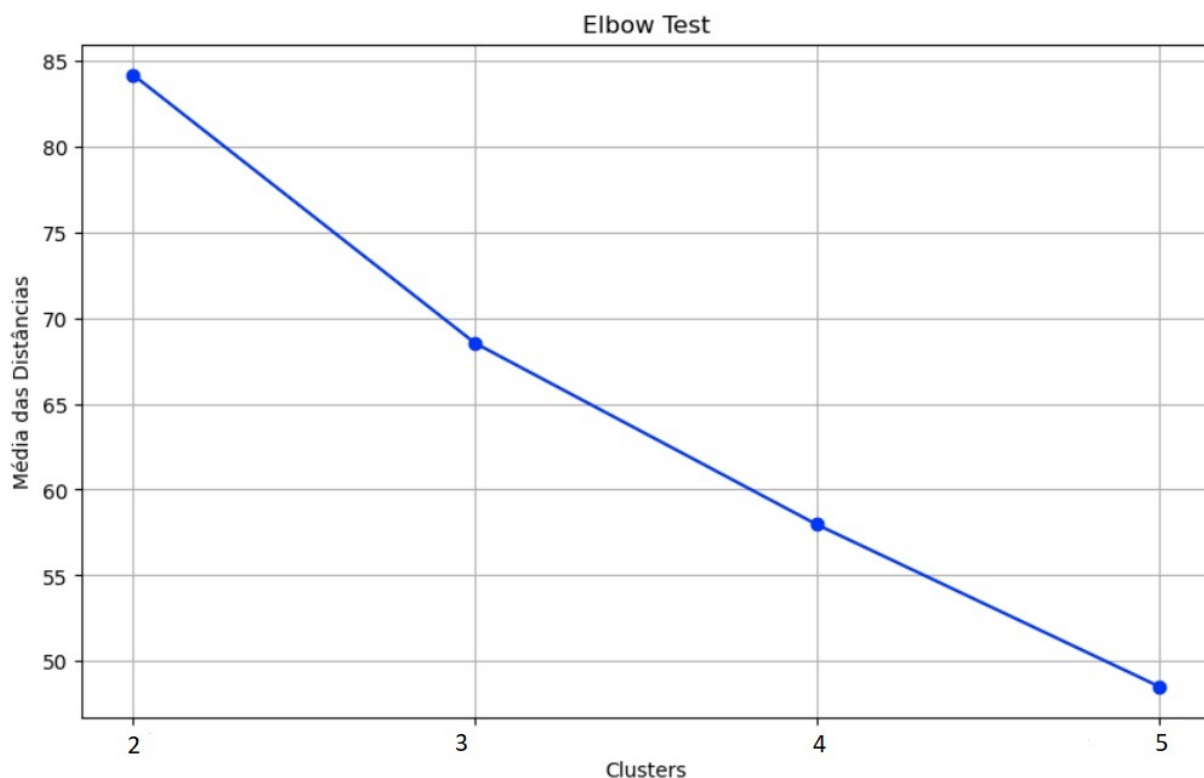


Figura 4: Exemplo do teste do cotovelo aplicado a um setor da Neoenergia Coelba. O teste indica que 3 clusters são ideais, pois é nesse ponto que ocorre a maior redução na média das distâncias dos pontos até o centro dos clusters.

## 2.5 O algoritmo Iterated Greedy

A maioria dos problemas no mundo real apresenta alta complexidade, frequentemente com restrições não-lineares e um amplo espaço de soluções. Isso exige o uso de técnicas capazes de resolver tais problemas em tempo hábil. Nesse contexto, as metaheurísticas surgiram como métodos de otimização que fornecem boas soluções em um tempo razoável, ao contrário dos métodos exatos, que obtêm soluções ótimas, mas muitas vezes impraticáveis.

A ideia principal da metaheurística LNS (Large Neighborhood Search) é melhorar gradualmente uma solução inicial, alternando entre a destruição e o reparo dessa solução. A heurística LNS pertence à classe de heurísticas conhecida como Very Large Scale Neighborhood search (VLSN) (Ahuja et al., 1998). Os algoritmos VLSN baseiam-se na observação de que uma pesquisa em uma vizinhança ampla permite encontrar ótimos locais de alta qualidade, resultando em soluções melhores. No entanto, pesquisar uma vizinhança ampla requer muito tempo. Para lidar com isso, várias técnicas de filtragem são usadas para limitar a busca. Nos algoritmos VLSN, a vizinhança é tipicamente restrita a um subconjunto de soluções que podem ser pesquisadas de forma eficiente. No LNS, a vizinhança é implicitamente definida por métodos (muitas vezes heurísticos) usados para destruir e reparar uma solução atual (Gendreau et al., 2010).

A metaheurística Iterated Greedy (IG) é um método simples para a resolução de problemas de otimização combinatória que utiliza a mesma ideia principal da LNS. O modelo IG gera uma sequência de soluções iterando sobre heurísticas gulosas construtivas, aplicadas a uma única solução, utilizando as duas fases principais da LNS: destruição e construção. Durante a fase de destruição, alguns componentes da solução são removidos, produzindo uma solução parcial. O procedimento de construção então aplica uma heurística construtiva gulosa para completar essa solução parcial. Em seguida, um critério de aceitação é aplicado para decidir se a nova solução candidata substituirá a solução atual. Opcionalmente, uma fase

de busca local pode ser adicionada para melhorar tanto a solução inicial quanto a solução reconstruída (García-Martínez et al., 2014). Uma característica do IG é a flexibilidade para incorporar componentes de outras metaheurísticas, com o objetivo de obter novos algoritmos híbridos mais eficazes (Lozano & García-Martínez, 2010; Talbi, 2002; Yuan et al., 2008).

O algoritmo proposto para resolver o problema é semelhante ao modelo de García-Martínez et al. 2014, porém adaptado para o problema da alocação de bases e modificado para incluir um critério de flexibilização. O algoritmo tem como entrada o número de bases propostas  $nb$ , o número de bases a serem removidas na fase de destruição  $d$ , o tabu tenure  $tt$  (número de iterações em que as bases desconstruídas não poderão participar da solução) e um critério de flexibilização que permite aceitar uma solução mesmo que não seja a melhor, o que amplia o espaço solução. Esse critério de flexibilização pode ser mais complexo, como os baseados na metaheurística Simulated Annealing (Ruiz & Stützle 2007) ou simplesmente um número aleatório entre 0 e 1, indicando a probabilidade de uma solução qualquer ser aceita pelo critério de flexibilização. O algoritmo minimiza a função objetivo  $f(S)$ , tendo como saída a melhor solução encontrada  $S_b$ .

Tal algoritmo foi aplicado e testado apenas para a Neoenergia Cosern. Antes da execução dele, são definidas as bases constantes, aquelas que já existem e estão em funcionamento, e as bases volantes, conjunto de municípios com mais de 10.000 habitantes que não fazem parte das bases constantes. Uma solução será sempre um subconjunto das bases volantes, porém para calcular o custo de uma determinada solução é necessário levar em conta também as bases constantes.

Em linhas gerais, o algoritmo funciona da seguinte forma: uma solução inicial  $S$  com  $nb$  bases é gerada seguindo um critério guloso, também são inicializadas a lista tabu e a lista de bases correntes e a melhor solução  $S_b$  como  $S$ . Para cada solução, são removidas aleatoriamente  $d$  bases na fase de destruição, estas são inseridas na lista Tabu e removidas da lista de bases correntes e não poderão entrar novamente na solução por algumas iterações. Em seguida, a solução é reconstruída usando as bases remanescentes da solução destruída, juntamente com novas bases da lista de bases correntes escolhidas com um critério guloso. Finalmente, se essa nova solução avaliada pela função objetivo  $f(S)$ , for melhor do que a original ela é aceita e se ainda for melhor do que a melhor solução passa a ser a melhor solução  $S_b$ .

A função objetivo  $f(S)$  representa o custo associado a cada solução. Esse custo é calculado utilizando uma matriz de custos que envolve 167 municípios e 74 bases candidatas. Para cada município  $i$  e base candidata  $j$ , consideram-se os seguintes fatores:

- $n_i$ : número de ocorrências ou clientes de um município;
- $m_i$ : distância mediana entre as ocorrências de um município e uma coordenada geográfica central que representa o município (geralmente próxima ao centro da cidade);
- $d_{ij}$ : distância da rota mais próxima entre o município e uma base candidata.

Cada valor na matriz de custos é calculado como  $n_i \times (d_{ij} + m_i)$ . A função  $f(S)$  é a soma dos menores custos para cada município em uma determinada solução.

O modelo utilizou a seguinte configuração padrão: número de iterações igual a 1000,  $nb$  igual a 5,  $d$  igual a 3,  $tt$  igual a 3 e  $flex$  igual a 0.5. No entanto, no modelo final, foram testados diferentes parâmetros, variando cada um deles e registrando os resultados. Os parâmetros testados foram:

- Número de bases adicionais : 5 a 10;
- Número de bases a serem removidas na etapa de destruição: 3 a 5;
- Tabu: 3 a 5;
- Critério de flexibilização: 0,5 e 0,8.

Isso resultou em um total de 108 soluções independentes. Os resultados foram agregados para calcular a probabilidade de uma base fazer parte da solução (figura 5).

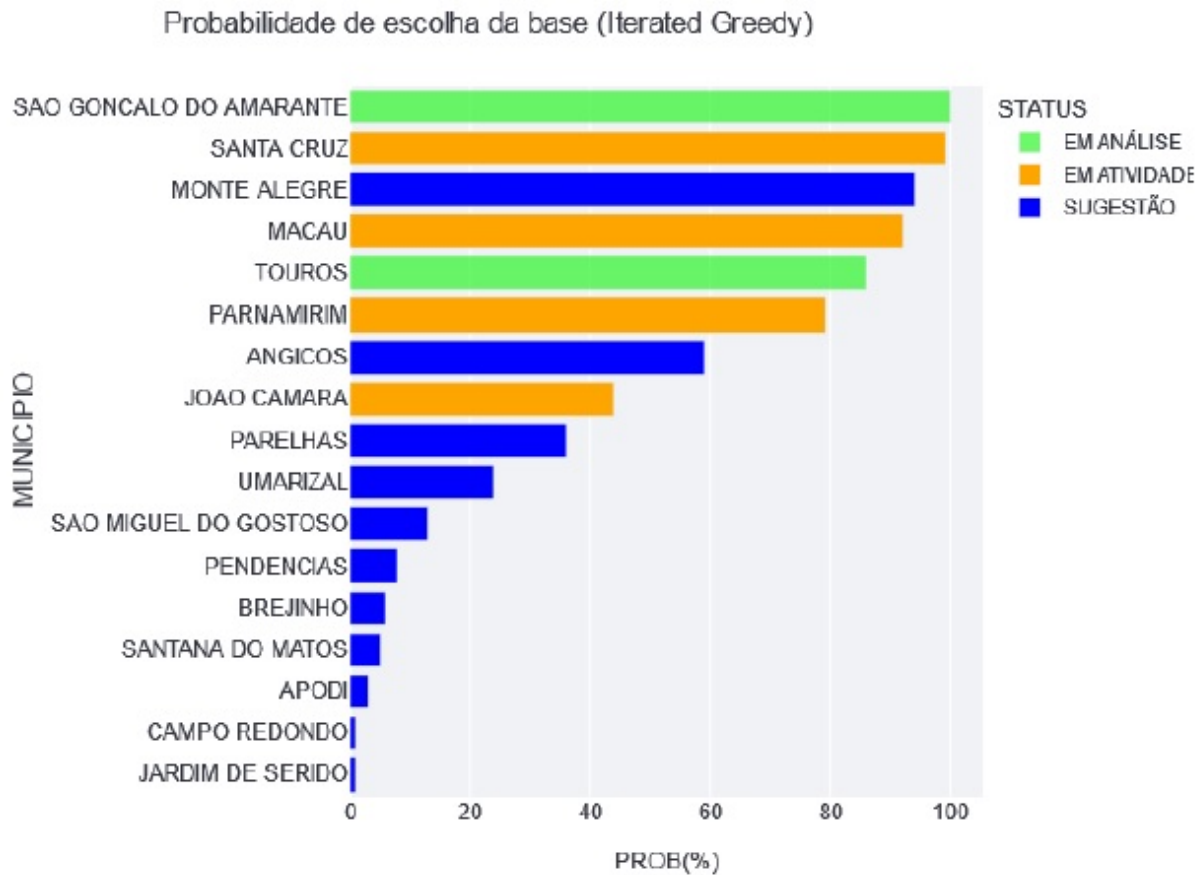


Figura 5: Probabilidade de escolha das bases na solução agregada do algoritmo Iterated Greedy. As barras verdes representam as bases sugeridas que estavam em análise para entrar em atividade. As barras laranjas indicam as bases sugeridas que já estão em atividade, mas que não fazem parte do dataset. A solução final do modelo consiste em um conjunto de nb bases propostas, com os municípios sendo atendidos tanto pelas bases atuais quanto pelas bases propostas. Com essas informações, é possível calcular as distâncias entre bases e ocorrências de forma semelhante ao modelo K-means. O modelo foi avaliado também por meio de um box-plot dessas distâncias, comparando-as com as distâncias atuais. Além disso, a posição geográfica das bases propostas e atuais é exibida em um mapa interativo do Rio Grande do Norte.



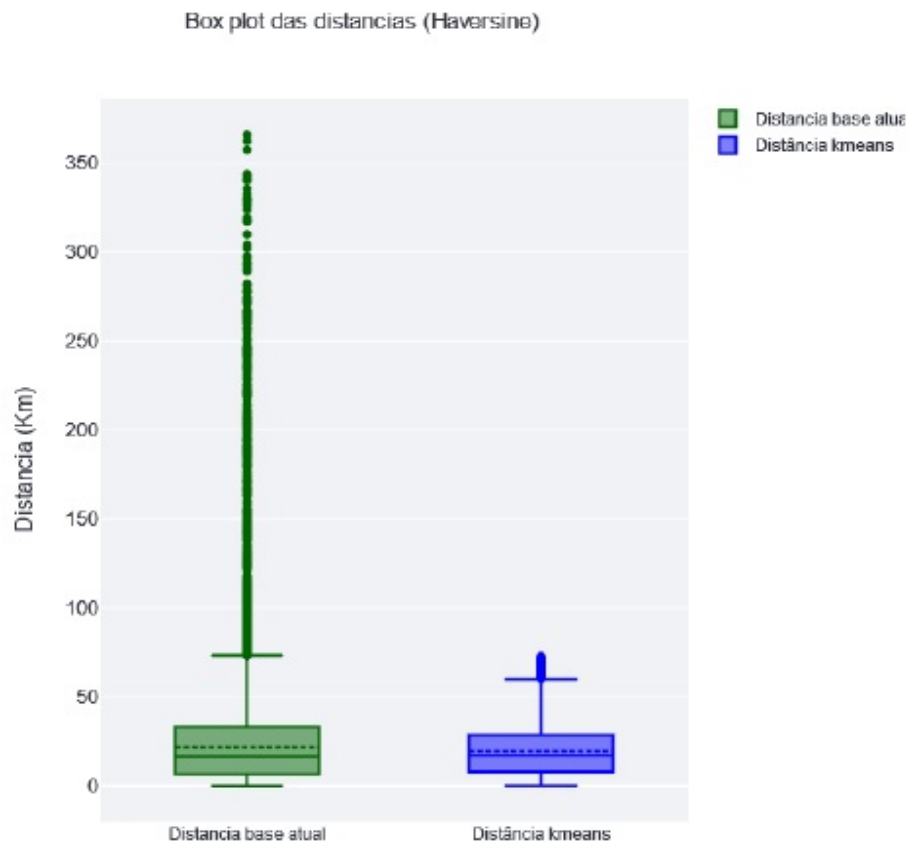


Figura 6: Exemplo de Box-plot para comparar as distâncias entre ocorrências emergenciais e bases. Foram utilizadas as distâncias de Haversine entre ocorrências não ponderadas e as bases antigas e propostas. A linha pontilhada indica a média.

## 2.6 Algoritmo para Otimização das Escalas de Trabalho

Otimizar os horários de entrada dos eletricitistas oferece diversas vantagens significativas para a gestão das escalas de trabalho. Essa otimização permite uma alocação mais eficiente dos recursos, garantindo que os eletricitistas estejam disponíveis nos momentos de maior demanda. Como resultado, o tempo de resposta a emergências pode ser reduzido, melhorando a satisfação dos clientes.

Nesse contexto, foi desenvolvido um algoritmo de otimização para determinar quantas equipes 5x2 e 6x3 são necessárias para atender toda a demanda, além de identificar os melhores horários de entrada para cada uma delas.

Os dados de entrada do algoritmo incluem o histórico do tempo de execução das demandas, somado ao tempo de deslocamento para cada hora de entrada das ocorrências. Esses tempos são calculados separadamente para fins de semana e dias úteis, considerando que as equipes 5x2 trabalham apenas nos dias úteis.

Com relação ao problema de otimização, temos que:

- Variáveis de decisão: Número de equipes em cada hora (dividido em dias úteis e fins de semana)
- Função Objetivo:

$$\min \sum_{i=1}^{48} (E_i - HH_i)^2 \quad (1)$$

onde

$i$  varia de 0 a 23 horas, para dias úteis e fins de semana

$E_i$  são os dados de entrada para cada hora, divididos em dias úteis e fins de semana

$HH_i$  são as horas trabalhadas se aquele horário tiver equipes disponíveis

- Restrições:

Número de equipes é um número inteiro e não-negativo

Toda a demanda do dia útil e fim de semana será finalizada ou também pode-se colocar restrições no número de equipes/número de veículos

Para realizar a otimização matemática mencionada, foi utilizado o Solver do Excel (ver figura 7), empregando o algoritmo GRG não linear (Generalized Reduced Gradient). O GRG opera de forma iterativa, ajustando as variáveis de decisão para minimizar ou maximizar a função objetivo, enquanto respeita as restrições impostas. Ele inicia com uma solução inicial e, a cada iteração, calcula a direção de descida mais promissora, ajustando as variáveis conforme necessário. Esse processo continua até que uma solução ótima seja encontrada ou que outros critérios de parada sejam atendidos, como um número máximo de iterações ou uma mudança mínima na função objetivo. O GRG é reconhecido por sua flexibilidade e capacidade de lidar com uma ampla variedade de problemas de otimização não linear, tornando-o uma ferramenta poderosa no Solver do Excel. Além disso, foram realizadas algumas simulações no Python (utilizando a biblioteca PuLP) e no Premium Solver do Excel.

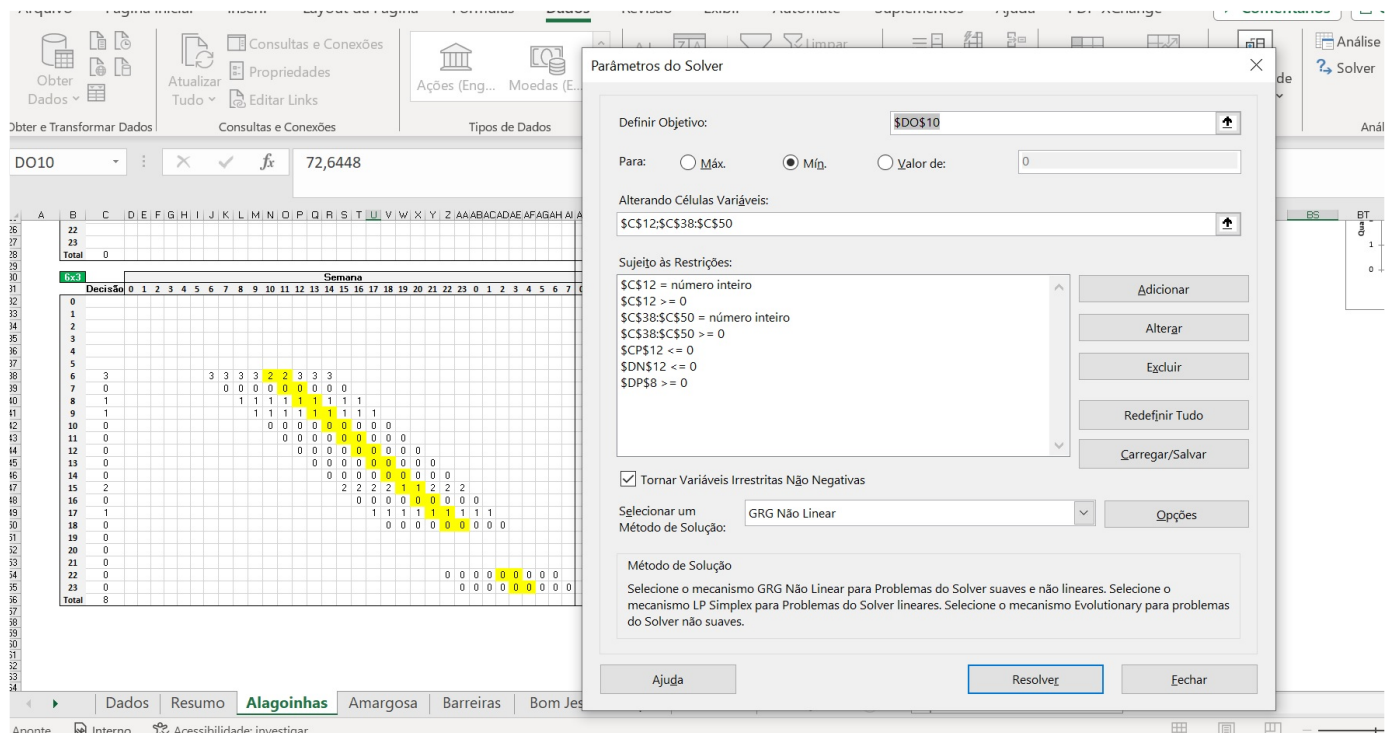


Figura 7: Tela do modelo de otimização de escalas no Solver do Excel.

## 2.7 Resultados

A abertura de novas bases reduz o primeiro deslocamento do eletricitista, pois esse é feito saindo diretamente da base para a ocorrência ou nota comercial. O primeiro deslocamento corresponde em cerca de

20% do deslocamento do dia para as empresas do grupo. Em geral, o foco é dado em bases com muitos atendimentos e primeiro deslocamento alto, como mostrado na figura 8.

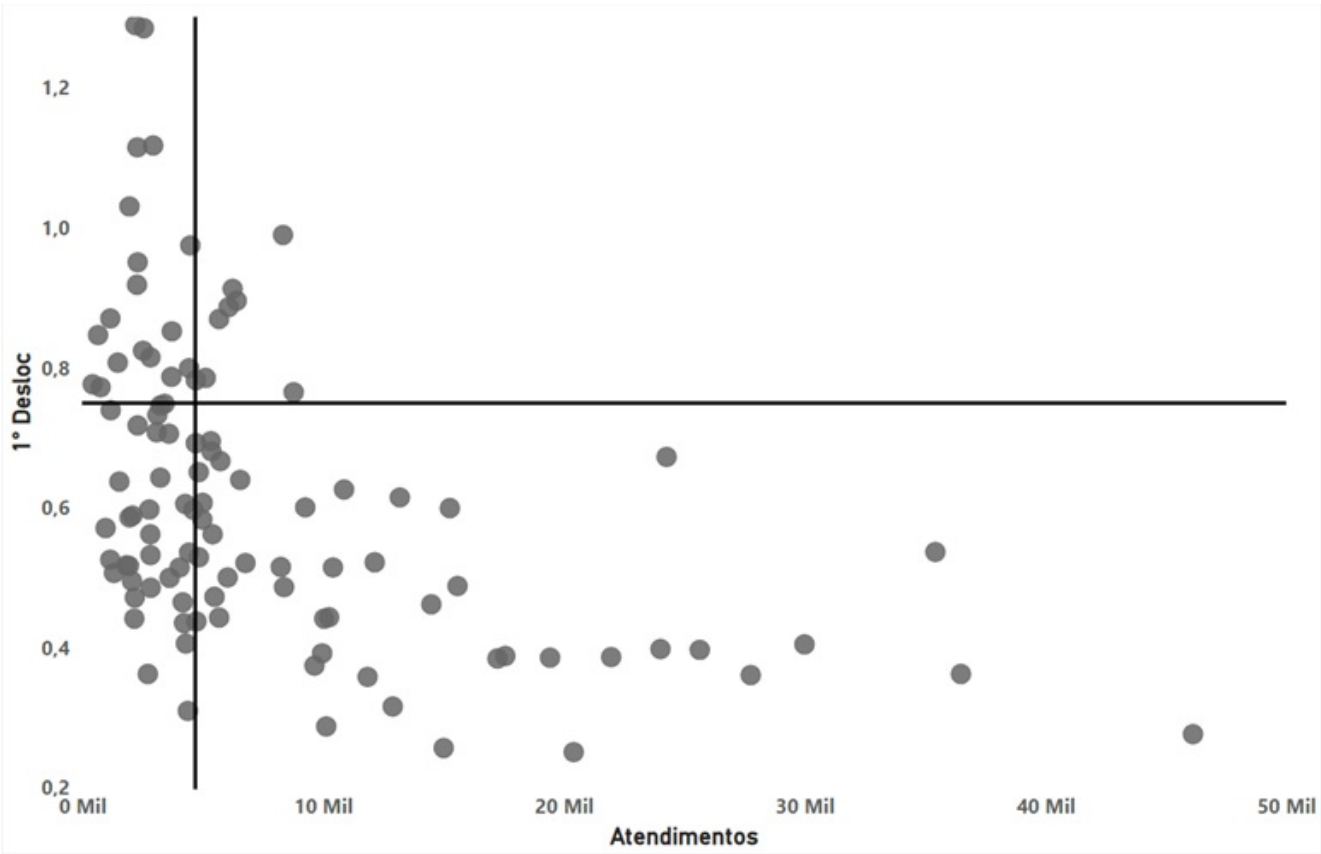


Figura 8: Gráfico de Atendimentos x Tempos de primeiro deslocamento por base. Em geral, são analisadas as bases com primeiro deslocamento alto e com muitos atendimentos.

A Tabela 1 mostra a quantidade de bases que foram e estão sendo abertas para cada uma das distribuidoras do grupo, bem como o ganho em equipes equivalentes decorrente da abertura dessas bases. Para as empresas do grupo, foram utilizados os algoritmos K-means clássico e K-means ponderado pelo CHI da ocorrência (cliente-hora interrompido) ou pela quantidade de clientes interrompidos da ocorrência. Apenas para a Cosern, foi executado o algoritmo iterated greedy. Após a execução dos algoritmos, é feita uma avaliação para verificar se o local indicado pelo algoritmo pode comportar uma base ou se é possível aproveitar algum local existente, como uma subestação ou outro prédio da companhia, para ser uma nova base.

Tabela 1: Quantidade de bases que foram ou serão abertas e ganho em equipes equivalentes por empresa

<b>Empresa</b>	<b>Qtde Bases</b>	<b>Ganho Equipes Equivalentes</b>
<b>Coelba</b>	20	10
<b>Elektro</b>	9	3
<b>Pernambuco</b>	28	21
<b>Cosern</b>	3	2
<b>Brasília</b>	4	3

Para todas as rodadas do algoritmo, são comparadas as distâncias atuais das bases com as ocorrências e/ou notas comerciais com as distâncias propostas pelos algoritmos (figura 6). Para se calcular essas distâncias, utiliza-se o pacote do python Osmnx e também se utiliza a api do Google maps. A figura 9 mostra as fotos de bases que foram planejadas após a execução dos algoritmos e a figura 10 ilustra um exemplo de rodada do algoritmo k-means clássico.

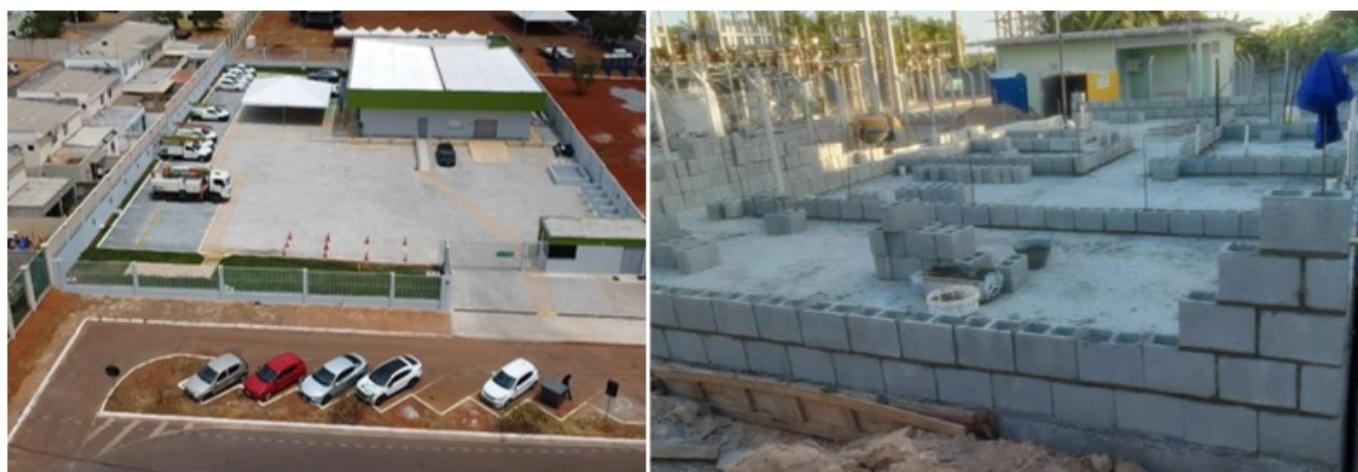


Figura 9: A foto à esquerda mostra a inauguração da base de São Sebastião, em Brasília. A foto à direita ilustra a construção da base de Apodi, na Cosern.

Em relação ao algoritmo de escalas, foi realizado um piloto na Coelba em duas bases operacionais (Iramaia e Itaberaba) durante o ano de 2024. Ambas as bases possuem apenas um veículo disponível, e os horários de entrada das equipes são às 7h e às 16h. O algoritmo sugeriu adiantar em uma hora a entrada das equipes, passando para às 6h e 15h.

No piloto, estimou-se uma redução do atendimento das ocorrências devido a essa mudança no horário de entrada, resultando em uma estimativa de redução do DEC de 1,7%.

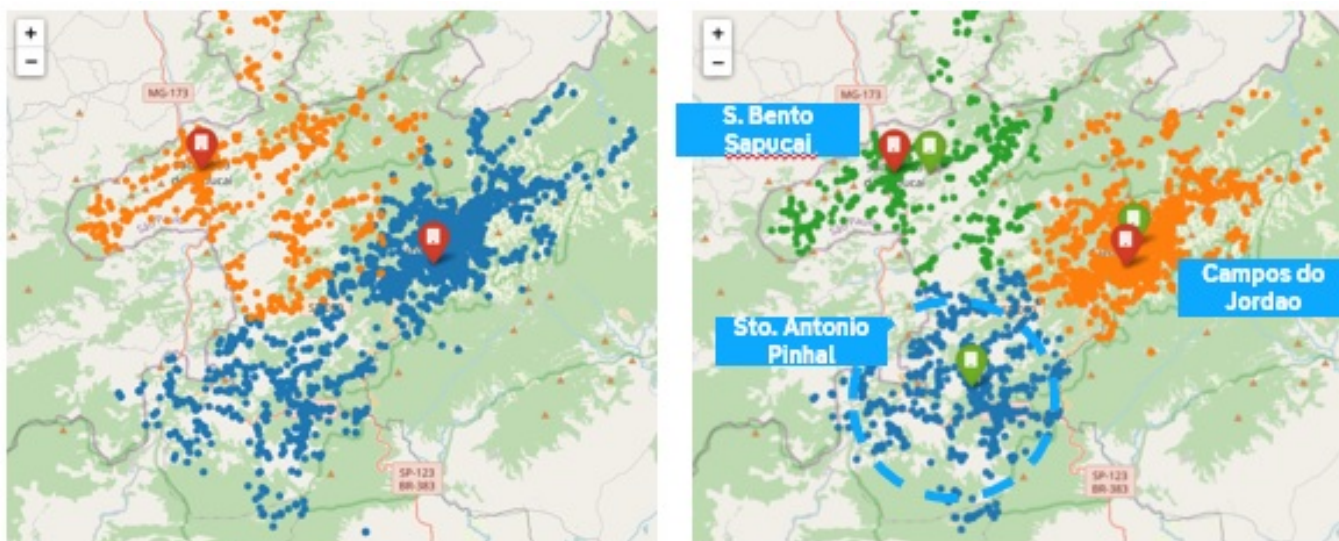


Figura 10: O mapa à esquerda mostra a situação antes da UTD (Unidade Territorial de Distribuição) de Campos do Jordão da Elektro. O mapa à direita mostra um exemplo da rodada do algoritmo kmeans clássico, sugerindo uma nova base no município de Santo Antônio do Pinhal.

### 3. Conclusão

Os algoritmos K-means clássico, Weighted K-means e o algoritmo de otimização iterated greedy mostraram-se bastante eficazes na determinação de uma boa alocação das bases. Da mesma forma, o algoritmo de otimização GRG demonstrou eficiência na alocação dos horários de entrada dos eletricitas. Com a abertura das novas bases, observou-se uma queda significativa nos primeiros deslocamentos. Além disso, com o algoritmo de otimização nas escalas, estimou-se uma redução no DEC (duração equivalente de interrupção por unidade consumidora) devido à melhor alocação das equipes de acordo com os horários de entrada das ocorrências.

Para trabalhos futuros, seria ideal trabalhar com projeção horária e espacial das ocorrências, considerando que apenas dados históricos foram utilizados para rodar os algoritmos. Para mitigar esse problema, em algumas ocasiões, os algoritmos foram executados com várias partições de dados ao longo do tempo, a fim de verificar se houve mudanças significativas em cada uma das rodadas. Outros trabalhos futuros poderiam incluir a execução de diferentes algoritmos de cluster para a alocação das bases, bem como métodos variados para avaliação da qualidade da clusterização. Outro ponto importante é que os algoritmos de clusterização calculam as distâncias euclidianas. O ideal seria desenvolver um algoritmo de clusterização baseado em distâncias rodoviárias, dado que as distâncias rodoviárias atualmente só são calculadas após a clusterização. Além disso, o algoritmo iterated greedy foi rodado apenas para a Cosern. A ideia é aplicá-lo também para as outras distribuidoras.

### 4. Referências bibliográficas



- Ahuja, R, J Orlin & D Sharma (1998), 'New neighborhood search structures for the capacitated minimum spanning tree problem, sloan school of management'.
- Alloghani, Mohamed, Dhiya Al-Jumeily, Jamila Mustafina, Abir Hussain & Ahmed J. Aljaaf (2020), A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science, Springer International Publishing, Cham, pp. 3–21.
- García-Martínez, Carlos, Francisco J Rodríguez & Manuel Lozano (2014), 'Tabu-enhanced iterated greedy algorithm: a case study in the quadratic multiple knapsack problem', *European Journal of Operational Research* 232(3), 454–463.
- Gendreau, Michel, Jean-Yves Potvin et al. (2010), *Handbook of metaheuristics*, Vol. 2, Springer.
- Johnson, R.; Wichern, D. *Applied Multivariate Statistical Analysis*. 6th ed. Pearson Prentice Hall, New Jersey, 2007.
- Lasdon, L. S., Fox, R. L., & Ratner, M. W. (1974). Nonlinear optimization using the generalized reduced gradient method *Revue Française d'Automatique, Informatique et Recherche Opérationnelle*, 8(3), 73-1031.
- Monard, Maria Carolina & José Augusto Baranauskas (2003), 'Conceitos sobre aprendizado de máquina', *Sistemas inteligentes-Fundamentos e aplicações* 1(1), 32.
- Ruiz, Rubén & Thomas Stützle (2007), 'A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem', *European journal of operational research* .
- Sandhu, Tejenderkaur H (2018), 'Machine learning and natural language processing- a review.', *International Journal of Advanced Research in Computer Science*
- Zhang, Aston, Zachary C Lipton, Mu Li & Alexander J Smola (2021), 'Dive into deep learning'