



UNIVERSIDADE FEDERAL DE MINAS GERAIS

GRADUAÇÃO EM ENGENHARIA DE SISTEMAS

**HENRIQUE DARÉ PERIM
VINICIUS CARDOSO ANTUNES**

**TRABALHO COMPUTACIONAL 1:
WLAN**

**Belo Horizonte, MG
2023**

SUMÁRIO

1	INTRODUÇÃO	2
2	METODOLOGIA	3
2.1	Parâmetros	3
2.2	Restrições	3
2.3	Distribuição de clientes	4
2.4	Função f_1	4
2.4.1	Definição do problema	4
2.4.2	Solução inicial	5
2.4.3	Estruturas de vizinhança	5
2.4.4	Refinamento local (busca local)	5
2.5	Função f_2	6
2.5.1	Definição do problema	6
2.5.2	Solução inicial	7
2.5.3	Estruturas de vizinhança	7
2.5.4	Refinamento local (busca local)	7
3	RESULTADOS E DISCUSSÕES	8
3.1	Soluções	8
3.2	Desempenho	10
4	CONCLUSÕES	11

1 INTRODUÇÃO

O objetivo deste trabalho é projetar e implementar uma rede WLAN (Wireless Local Area Network) do tipo N 2D para atender a um centro de convenções com dimensões de 400 x 400 metros. O planejamento da rede leva em consideração a presença de 256 pontos de demanda, cada um com suas coordenadas geográficas e requisitos de largura de banda individuais.

Para a realização desse projeto, o arquivo "clientes.csv" foi utilizado, contendo informações detalhadas sobre os clientes. Cada linha do arquivo representa um cliente, sendo as duas primeiras colunas dedicadas às coordenadas x e y da localização do cliente em metros, enquanto a terceira coluna indica o consumo de largura de banda necessário para cada cliente, medido em Mbps.

O trabalho visa criar uma infraestrutura de rede eficiente e confiável que seja capaz de atender às necessidades de conectividade dos 256 pontos de demanda, garantindo uma cobertura adequada e a distribuição eficaz dos recursos de largura de banda. A abordagem adotada levará em conta a localização estratégica dos clientes e suas demandas individuais, proporcionando uma solução de rede personalizada e otimizada para o centro de convenções.

2 METODOLOGIA

2.1 Parâmetros

- n : nr de clientes
- m : nr de possíveis locais para o PA
- c_j : consumo do cliente j
- q_i : capacidade do PA i
- r_i : raio de cobertura do PA i
- η : taxa de cobertura dos clientes
- d_{ij} : distância entre o PA i e o cliente j
- m_{max} : quantidade de PAs disponíveis

2.2 Restrições

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} \geq n\eta \quad (R1) \quad (1)$$

$$\sum_{j=1}^n c_j x_{ij} \leq y_i q_i, \forall i \in \{1, \dots, m\} \quad (R2) \quad (2)$$

$$d_{ij} x_{ij} \leq y_i r_i, \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, n\} \quad (R3) \quad (3)$$

$$\sum_{i=1}^m x_{ij} \leq 1, \forall j \in \{1, \dots, n\} \quad (R4) \quad (4)$$

$$\sum_{i=1}^m y_i \leq m_{max} \quad (R5) \quad (5)$$

$$x_{ij} \in \{0, 1\}, \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, n\} \quad (R6) \quad (6)$$

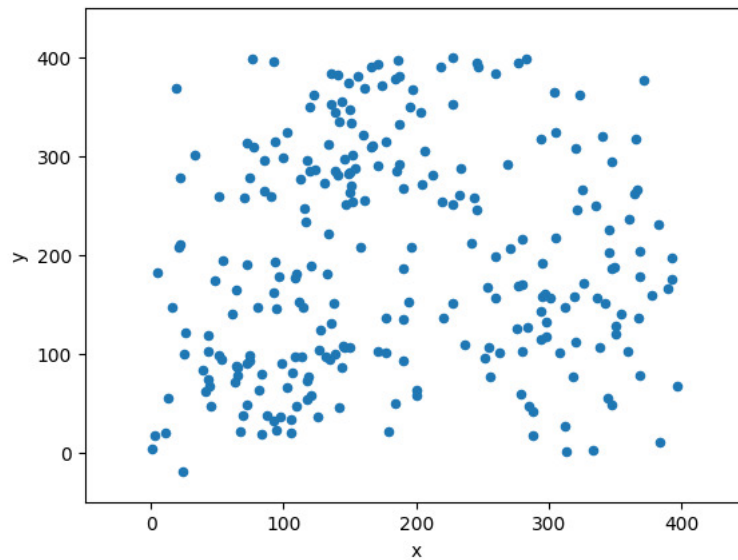
$$y_i \in \{0, 1\}, \forall i \in \{1, \dots, m\} \quad (R7) \quad (7)$$

em que:

- R1 garante que o percentual mínimo de clientes seja atendido;
- R2 garante que a capacidade dos PAs ativos não seja violada;
- R3 garante que PAs ativos só atendam clientes que estejam dentro do seu raio de cobertura;
- R4 garante que cada cliente j seja atribuído a no máximo um PA;
- R5 garante que o número máximo de PAs ativos não seja violado;
- R6 e R7 definem o domínio das variáveis de otimização do problema.

2.3 Distribuição de clientes

Figura 1 – Distribuição de clientes no centro de convenções



Fonte: Autores.

2.4 Função f_1

Com o intuito de minimizar a quantidade de PAs ativos, pode-se escrever a seguinte função objetivo:

$$\min f_1 = \sum_{i=1}^m y_i$$

Onde y_i é 1 se o PA i está ativo e 0 caso contrário.

2.4.1 Definição do problema

Os PAs consistem em uma estrutura de dataframe do módulo pandas do Python, ou seja, uma matriz $3 \times n$ onde n é o número de PAs ativos. A primeira coluna é a coordenada x , a segunda coluna é a coordenada y e a terceira coluna é a quantidade de carga disponível da PA ativa.

Os clientes são representados por uma matriz 3×256 . A primeira coluna é a coordenada x , a segunda coluna é a coordenada y e a terceira coluna consiste de listas das PAs que estão mais próximas do cliente num raio de 70 m.

2.4.2 Solução inicial

Para este problema foi utilizado o kmeans para encontrar 25 centroides onde seriam alocados as primeiras PAs.

O algoritmo K-Means é uma técnica popular de análise de agrupamento (clustering) usada em aprendizado de máquina e mineração de dados. Seu principal objetivo é dividir um conjunto de dados em grupos ou clusters, onde os pontos de dados em cada cluster são semelhantes entre si e diferentes dos pontos em outros clusters. Essa técnica é útil para identificar padrões ocultos nos dados e agrupar elementos similares juntos, facilitando a análise e interpretação dos dados.

O funcionamento do K-Means pode ser resumido em etapas simples:

- Inicialização: O algoritmo começa com a seleção de K centroides iniciais (usamos 25), onde K é um número definido pelo usuário que representa o número desejado de clusters.
- Atribuição de pontos: Cada ponto de dados é atribuído ao cluster cujo centróide está mais próximo com base em alguma medida de distância, geralmente a distância euclidiana.
- Recálculo dos centroides: Os centroides de cada cluster são recalculados como a média dos pontos atribuídos a esse cluster. Esses novos centroides representam o "centro" atualizado de cada cluster.
- Reatribuição de pontos: Os pontos de dados são novamente atribuídos aos clusters com base nos centroides atualizados.
- Repetição: Os passos 3 e 4 são repetidos até que os centroides não mudem significativamente ou um critério de parada seja atendido, como um número máximo de iterações.

2.4.3 Estruturas de vizinhança

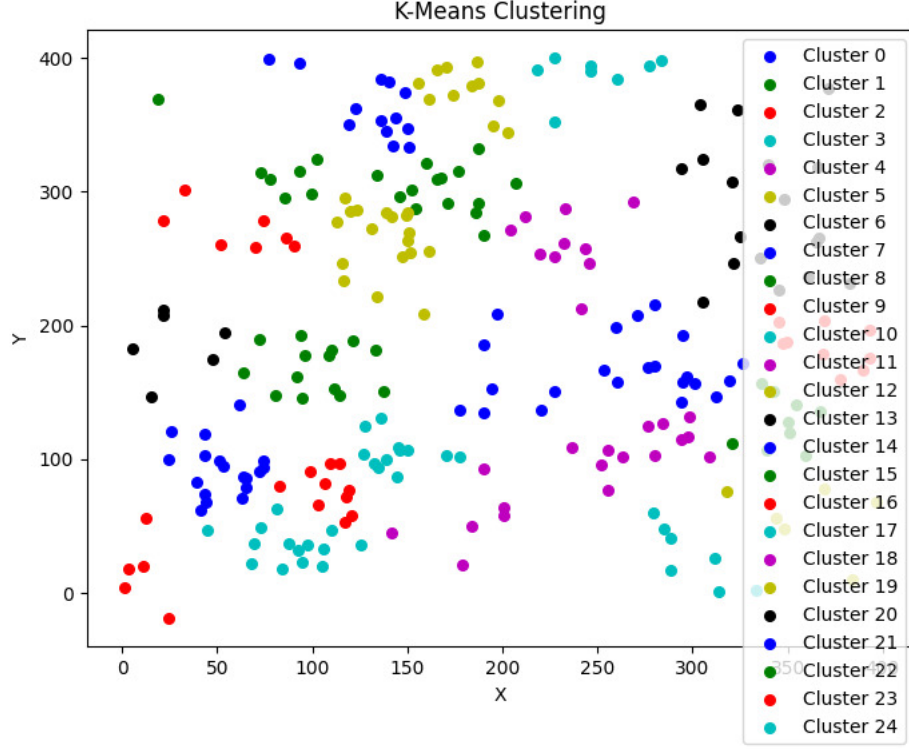
Para a função f_1 , foi pensando em 3 estruturas de vizinhança:

- Retira o PA menos requisitado
- Retira PA mais próximo de outro
- Retira os dois PAs mais próximos entre si e substitui pela média

2.4.4 Refinamento local (busca local)

Para a busca local, foi utilizado as estruturas de vizinhanças definidas e o kmeans para o neighborhood change.

Figura 2 – Clusterização dos clientes



Fonte: Autores.

2.5 Função f_2

Com o intuito de minimizar a soma das distâncias euclidianas entre os PAs e os clientes, pode-se escrever a seguinte função objetivo:

$$\min f_2 = \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij}$$

Onde x_{ij} é 1 se o cliente j é atendido pelo PA i e 0 caso contrário. d_{ij} representa a distância euclidiana entre o PA i e o cliente j .

2.5.1 Definição do problema

Para definir o problema para a função f_2 , foi criada uma função *probe f*, que recebe como argumento $m = 25$, que representa o número máximo de pontos de acesso possíveis e $n = 256$, que representa o número de clientes distribuídos no centro de convenções e o nome do arquivo que contém a tabela de informações dos clientes (coordenada x, coordenada y e consumo de banda), nomeado de *clientes.csv*.

Para restringir o problema, foram definidas variáveis responsáveis por limitar a taxa de cobertura dos clientes em 95%, limitar a cobertura de cada ponto de acesso em um raio de 70 metros e limitar a capacidade máxima dos pontos de acesso em 54 Mbps.

Após definir as variáveis, o arquivo que contém os dados do cliente foi lido e convertido em um dataframe utilizando a biblioteca *pandas* do Python. Percorrendo cada ponto de acesso e cada cliente, é possível calcular a distância euclidiana entre os pontos e clientes através do Teorema de Pitágoras e armazenar esses dados no objeto do problema, que será retornando no fim da função.

2.5.2 Solução inicial

A função *sol_inicial* é responsável por otimizar a alocação inicial de clientes aos PAs considerando as restrições apresentadas, como capacidade dos PAs, raio de cobertura e distância entre os clientes e os PAs.

A função inicia criando uma estrutura de dados para representar a solução, na qual cada PA é associado a uma lista que armazena os índices dos clientes alocados a esse PA. Essas listas são inicialmente vazias.

Em seguida, a função gera uma lista de clientes e a ordena com base na distância ao PA mais próximo. Essa ordenação visa alocar inicialmente os clientes mais próximos aos PAs disponíveis, buscando otimizar a eficiência da rede.

Para cada cliente na lista ordenada, a função verifica a capacidade disponível de cada PA dentro do raio de cobertura. Ela calcula a capacidade restante considerando os clientes já alocados a cada PA. A função, então, escolhe o PA mais próximo com capacidade disponível para alocar o cliente atual, considerando a distância entre o cliente e os PAs disponíveis, bem como a capacidade restante em cada PA.

O cliente é alocado ao PA escolhido, e o índice do cliente é adicionado à lista de clientes alocados a esse PA. A função retorna a estrutura de dados que representa a alocação inicial de clientes a PAs.

2.5.3 Estruturas de vizinhança

Para a função f_2 , foi pensando em 3 estruturas de vizinhança:

- Aproximar a PA com menor demanda em direção à PA mais próxima com maior demanda
- Aproximar a PA com maior demanda em direção à PA mais próxima com maior demanda
- Deslocar a PA que tenha boa parte de sua área de cobertura fora do plano definido (400 x 400) para mais dentro do grid

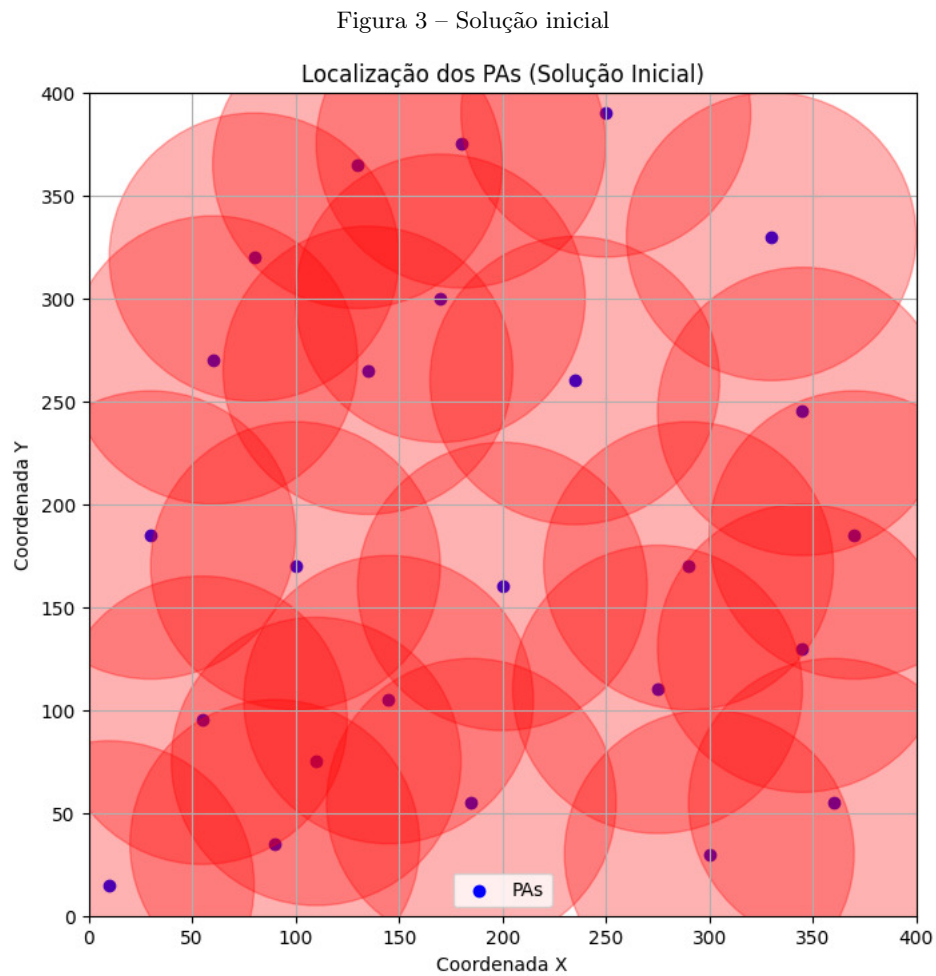
2.5.4 Refinamento local (busca local)

Para a busca local, foi utilizado as estruturas de vizinhanças definidas e o kmeans para o neighborhood change.

3 RESULTADOS E DISCUSSÕES

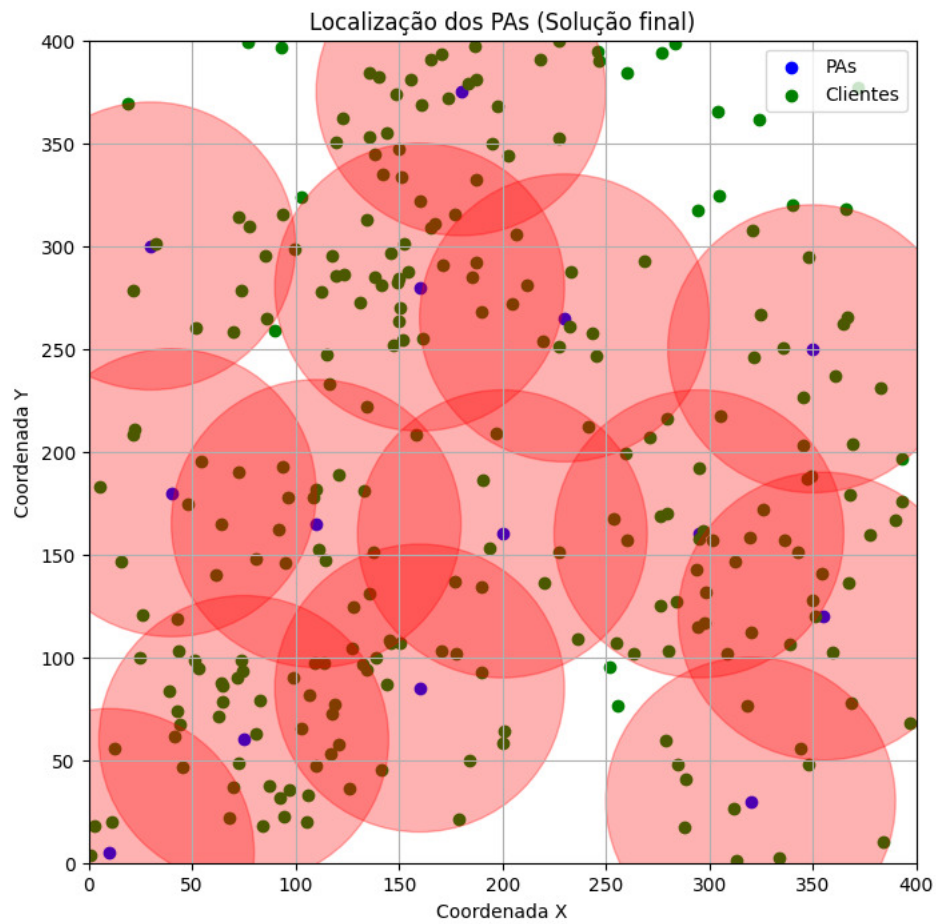
3.1 Soluções

Após executar o algoritmo na função 1, foram obtidos os seguintes resultados:



Fonte: Autores.

Figura 4 – Solução final

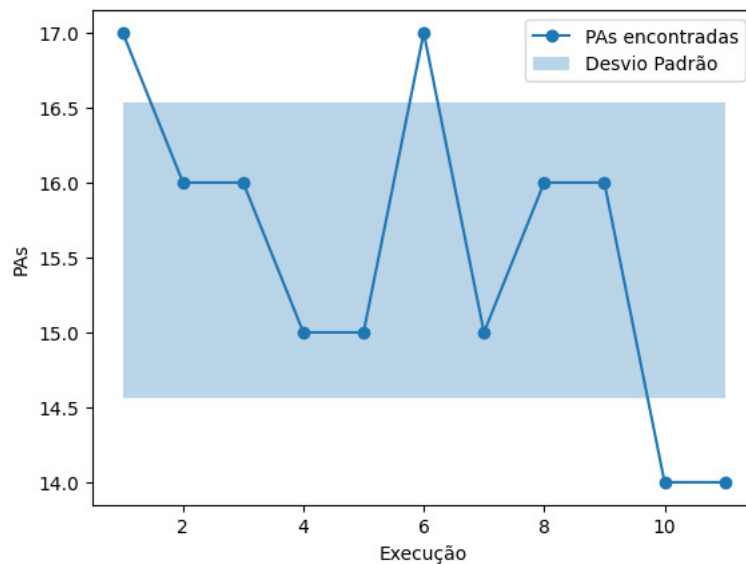


Fonte: Autores.

Pode-se perceber que o número de PAs diminui consideravelmente, porém a abrangência dos clientes não foi alterada de maneira significativa, gerando uma boa solução.

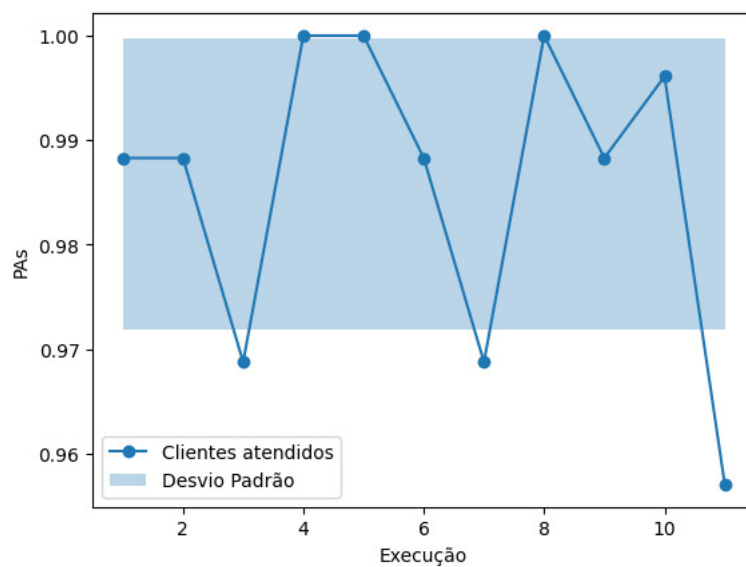
3.2 Desempenho

Figura 5 – Quantidade de PAs necessários



Fonte: Autores.

Figura 6 – Cobertura de clientes



Fonte: Autores.

Nas imagens acima temos a quantidade de PAs mínimo encontrado e a cobertura de clientes encontrada após cada execução do algoritmo.

Pode-se perceber que a penúltima execução foi a melhor das soluções encontradas, pois possui o menor número de PAs e a maior área de cobertura.

4 CONCLUSÕES

Pode-se perceber que como se trata de um método estocástico, provavelmente não obteremos o mesmo resultado de outra execução. A minimização foi executada 10 vezes, sendo possível encontrar uma boa solução com 14 PAs e cerca de 99% de cobertura dos clientes.

Provavelmente não é a melhor solução, mas em tempo hábil esta solução se mostra interessante. Também se apresenta melhor do que a da solução inicial, que por si só já entendia as restrições. Diante disso, é concluído que a heurística construtiva, junto com as estruturas de vizinhança servem para o nosso problema, possuindo um papel fundamental na otimização da situação e permitindo uma redução de custos e melhor aproveitamento dos recursos disponíveis.