

# Trabalho Computacional - Teoria da Decisão

Pedro Loes

17/01/2021

---

## Abstract

- This article comprises an applied problem of the Decision Theory course from the department of electrical engineering of UFMG taught by Professor Lucas Batista. The objective of this work is to apply the optimization and decision techniques taught throughout the semester. The proposed problem aims to minimize the number of access points routers to supply the Mbps consumption demanded by clients in a controlled environment. Also meeting pre-established optimization restrictions.

## Introdução

- O problema foi modelado no software R com implementação da função objetivo em C++.
- Diferentes espaços de busca foram considerados com o intuito de descobrir a partir de qual densidade de espaço de busca discretizado o algoritmo convergiria para a solução de mínimo global ótima.

## Especificações do Problema

- Deseja-se instalar uma rede WLAN do tipo N 2D para atendimento de um centro de convenções com  $800 \times 800$  metros. Para planejamento dessa rede foram estimados 500 pontos de demanda, com suas respectivas posições geográficas e consumos de largura de banda. O arquivo clientes.csv contém:
  - Coordenada x do cliente em metros.
  - Coordenada y do cliente em metros.
  - Consumo de banda do cliente em Mbps.
- Amostra das 10 primeiras observações:

Id	x	y	Mbps
1	168.99	220.62	0.47347
2	218.11	211.16	0.41026
3	207.48	157.38	1.37570
4	213.25	189.93	0.93025
5	199.19	140.71	0.15790
6	177.74	188.51	0.94899
7	197.44	176.30	0.30420
8	161.48	197.12	1.16990
9	189.84	211.32	1.29950
10	269.22	224.90	0.51370

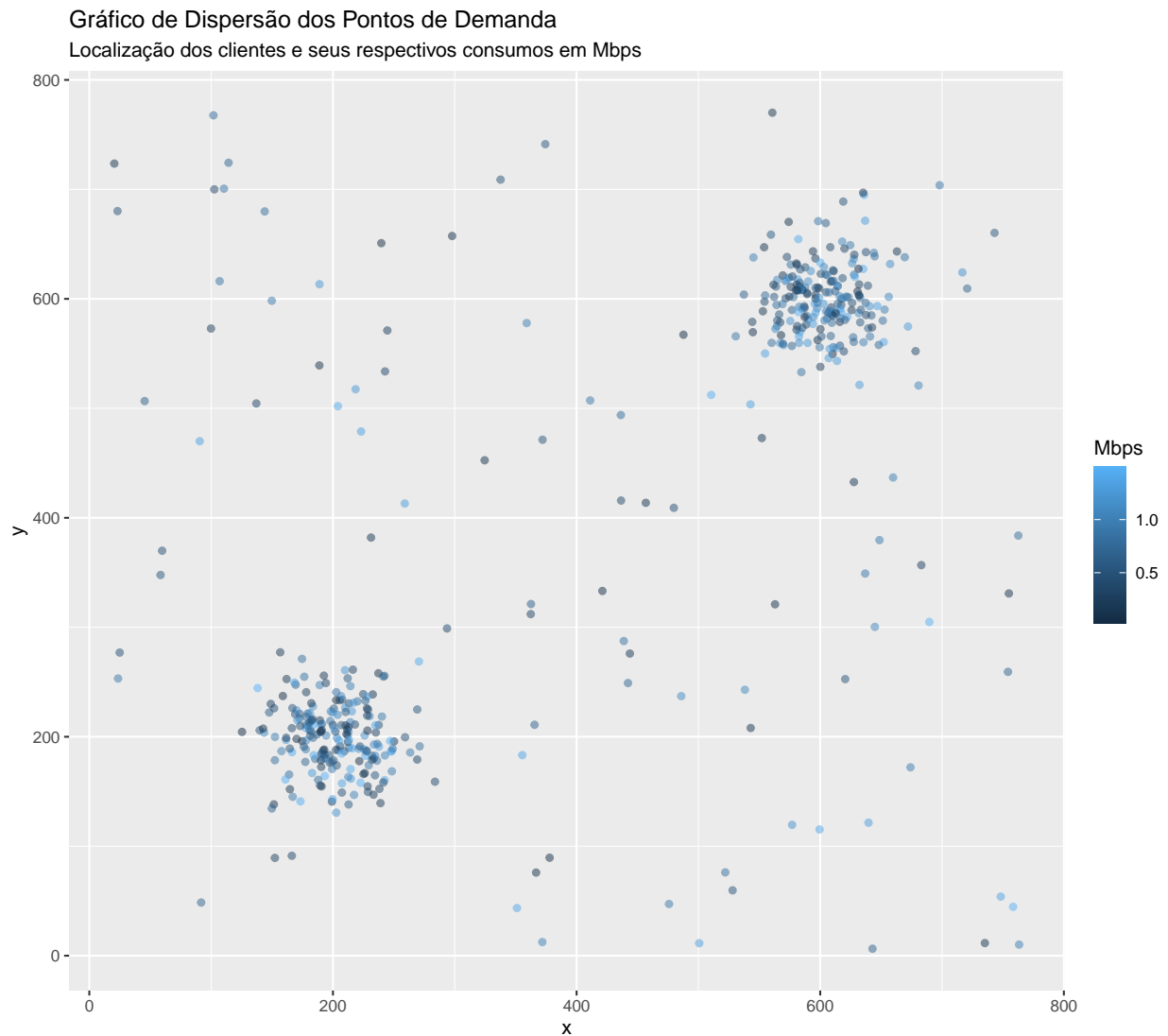
- Variáveis de Decisão:
    - Coordenadas dos pontos a serem instalados na área de 800 x 800 metros.
    - Ponto de Acesso que será responsável pelo atendimento de cada cliente.
  - Restrições:
    - Ao menos 95% dos pontos de demanda devem ter suas demandas integralmente atendidas.
    - Cada ponto de acesso a ser instalado tem capacidade de 150Mbps, que não pode ser excedida.
    - Um cliente pode ser atendido por um PA se a distância entre ambos é inferior a 85 metros.
    - Cada cliente só pode ser atendido por um único PA.
    - Devido a restrições orçamentárias, podem ser instalados no máximo 100 PA's.
  - Simplificações:
    - Os pontos de demanda e seus consumos de banda são estáticos.
    - O efeito de obstáculos no ambiente são desprezados.
    - Um ponto de acesso não causa interferência em outros.
- 

## Modelagem

- Definição de Variáveis:
    - $Pd_i \leftarrow$  Pontos de Demanda  $i = \{1, 2, 3, \dots, 500\}$ ,  $Pd_i \in R\{x, y\}$ ,  $x, y = \{1, \dots, 800\}$
    - $Pa_j \leftarrow$  Pontos de Acesso  $j = \{1, 2, \dots, 100\}$ ,  $Pa_j \in R\{x, y\}$ ,  $x, y = \{1, \dots, 800\}$
    - $\beta_i \leftarrow$  Consumo de banda em Mbps dos Pd's,  $\forall i \in Pd_i$
    - $d_{i,j} \leftarrow$  Distância Euclidiana entre  $Pd_i$  e  $Pa_j$ ,  $\forall i \in Pd_i, \forall j \in Pa_j$
    - $\alpha_j \leftarrow$  Ativação do  $Pa_j$ ,  $\alpha_j \in \{0, 1\} \forall j \in Pa_j$
    - $\eta_{i,j} \leftarrow$  Atendimento do  $Pd_i$  pelo  $Pa_j$ ,  $\eta_{i,j} \in \{0, 1\} \forall i \in Pd_i, \forall j \in Pa_j$
  - Definição da Função 1:
    - $\min (\sum_{j=1}^{|Pa|} \alpha_j)$
  - Definição da Função 2:
    - $\min (\sum_{i=1}^{|Pd|} \sum_{j=1}^{|Pa|} d_{i,j} \times \eta_{i,j})$
  - Definição das Restrições:
    - $\sum_{i=1}^{|Pd|} \sum_{j=1}^{|Pa|} \eta_{i,j} \geq 475$
    - $\sum_{i=1}^{|Pd|} \eta_{i,j} \times \beta_i \leq 150 \quad \forall j \in \alpha_j = 1$
    - $\sum_{j=1}^{|Pa|} \eta_{i,j} \leq 1 \quad \forall i \in Pd$
    - $\sum_{j=1}^{|Pa|} \alpha_j \leq 100$
    - $\exists \eta_{i,j} \forall d_{i,j} \leq 85$
-

## Análise Exploratória

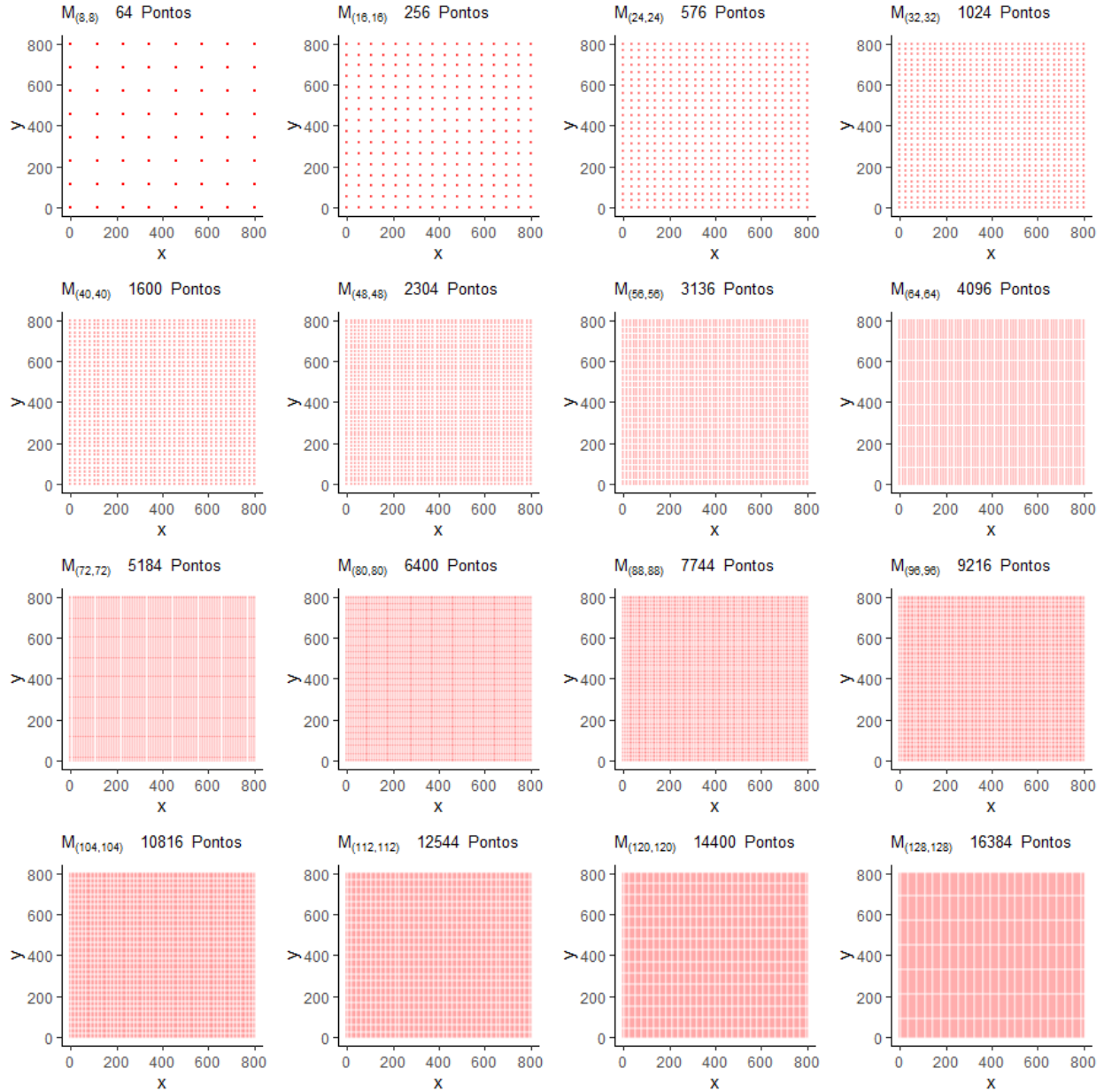
- Para ilustrar uma visão inicial da distribuição dos clientes no espaço de 800 x 800 foi elaborado um gráfico de dispersão 2d com os dados da localização (x, y) dos pontos de demanda.
- O parâmetro alpha de transparência foi alterado para 0.5 com o intuito de eliminar o problema da sobreposição de Pd's e facilitar a compreensão da densidade destes pontos.
- A dimensão de cor foi adicionada considerando o gradiente das cores preta até azul para o consumo de largura de banda requisitada em Mbps por cada cliente.



- Existem 2 grandes clusters na dispersão da localização dos clientes em torno dos pontos (200, 200) e (600, 600). Os demais pontos apresentam distribuição esparsa na área de 800 x 800.
- Na diagonal que vai de (0,0) até (800,800) os pontos de demanda são menos espalhados enquanto que na diagonal (0,800) até (800,0) os pontos estão mais dispersos.
- Parece não existir padrão visual para a distribuição das demandas de Mbps.

## Espaços de Busca

- A construção de empaços de busca considerou espaços discretizados na forma de matrizes quadradas do tipo  $M_{(i,j)}$  com  $i = \{8, 9, \dots, 150\}$  e  $j = \{8, 9, \dots, 150\}$  dentro da área de 800 x 800. Cada  $(i,j)$  representa a possível localização de um ponto de acesso.
- Para ilustrar a construção do espaço discretizado de busca foram consideradas as matrizes de  $M_{8,8}$  até  $M_{128,128}$ .



- Conforme a dimensão da matriz de espaço de busca aumenta é possível visualizar o aumento de granularidade na inspeção do espaço de busca.
- O algoritmo completo considerou espaços discretos entre 8 x 8 até 150 x 150.
- [LINK da Apresentação em Slides: Como Funciona o Algoritmo](#)

## Prototipagem do Algoritmo

- Função Conta Pontos em C++

```
double square(double x){double squared = x * x; return squared;}
std::vector<double> conta_pontos(x, y, wlanx, wlany, indice) {
    int n = wlany.size(); double soma; double raio = square(85); indices(0);
    for(int i = 0; i < n; i++) {
        soma = square(wlanx[i] - x) + square(wlany[i] - y);
        if( soma <= raio) {
            indices.push_back(indice[i]);}}
    return indices;
}
```

- Algoritmo de Convergência em R

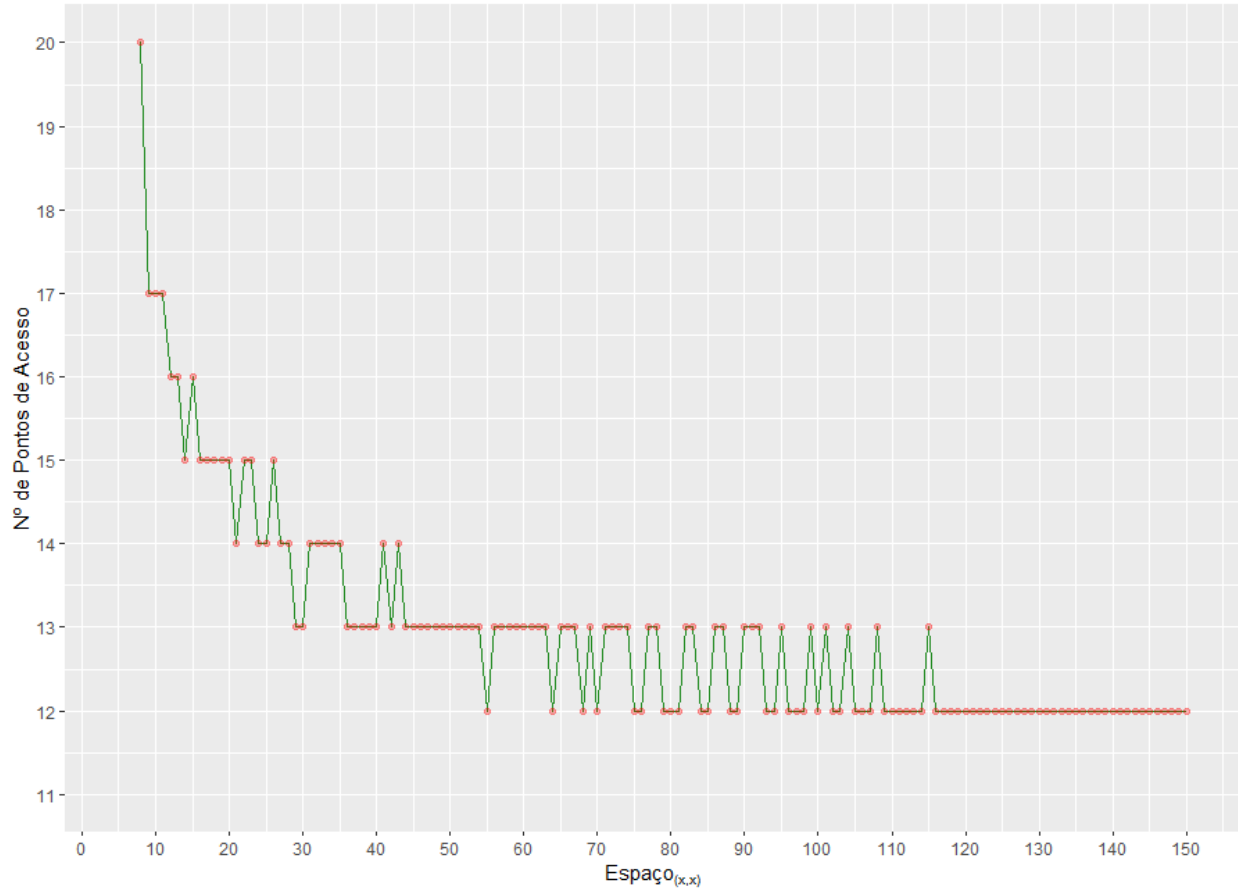
```
wlan_completa <- read_csv("clientes.csv", col_names = c("x", "y", "Mbps"))
wlan_id <- wlan_completa %>% mutate(indice = 1:nrow(wlan_completa))
sourceCpp('conta_pontos.cpp')
performance <- tibble(intervalos = 8:50, PA = rep(0,143), tempo = rep(0,143))
for(intervalo in seq(8, 150, by = 1)){
    wlan <- wlan_id
    centro_x <- rep(seq(0, 800, length.out = intervalo), each = intervalo)
    centro_y <- rep(seq(0, 800, length.out = intervalo), times = intervalo)
    vencedores <- NULL; contador <- 0 ;Mbps <- NULL; viola <- NULL
    inicio <- Sys.time()
    while(contador < 475){
        cobertura <- NULL
        populoso <- 1
        for(i in seq(2, intervalo^2, by = 1)){
            if(length(conta_pontos(x = centro_x[i], y = centro_y[i],
                                   wlanx = wlan$x, wlany = wlan$y, indice = wlan$indice)) >=
                length(conta_pontos(x = centro_x[populoso], y = centro_y[populoso],
                                   wlanx = wlan$x, wlany = wlan$y, indice = wlan$indice))&
                (!i %in% viola) ){
                populoso <- i
                cobertura <- conta_pontos(x = centro_x[populoso], y = centro_y[populoso],
                                           wlanx = wlan$x, wlany = wlan$y, indice = wlan$indice)}}
        megas <- wlan %>%
            filter(indice %in% as.integer(cobertura)) %>%
            summarise(total = sum(Mbps)) %>%
            pull(total)
        if(megas < 150){
            Mbps <- c(Mbps, megas)
            vencedores <- c(vencedores, populoso)
            wlan <- wlan %>% filter(!indice %in% as.integer(cobertura) )
            contador <- contador + length(cobertura)} else { viola <- c(viola, populoso)}}
    fim <- Sys.time()
    performance$tempo[intervalo-7] <- fim - inicio
    performance$PA[intervalo-7] <- length(vencedores)
}
```

## Convergência

- O algoritmo completo utilizado para convergência considerou espaços de busca de  $M_{8,8}$  até  $M_{150,150}$  contabilizando o máximo de 22500 coordenadas para possíveis localizações dos pontos de acesso.
- Para cada espaço de busca foi contabilizado o número de Pontos de Acesso para atingir uma solução ótima.

### Convergência do N° de Pontos de Acesso

Quantidade mínima de pontos de acesso por espaço de busca que atendem os requisitos do problema.



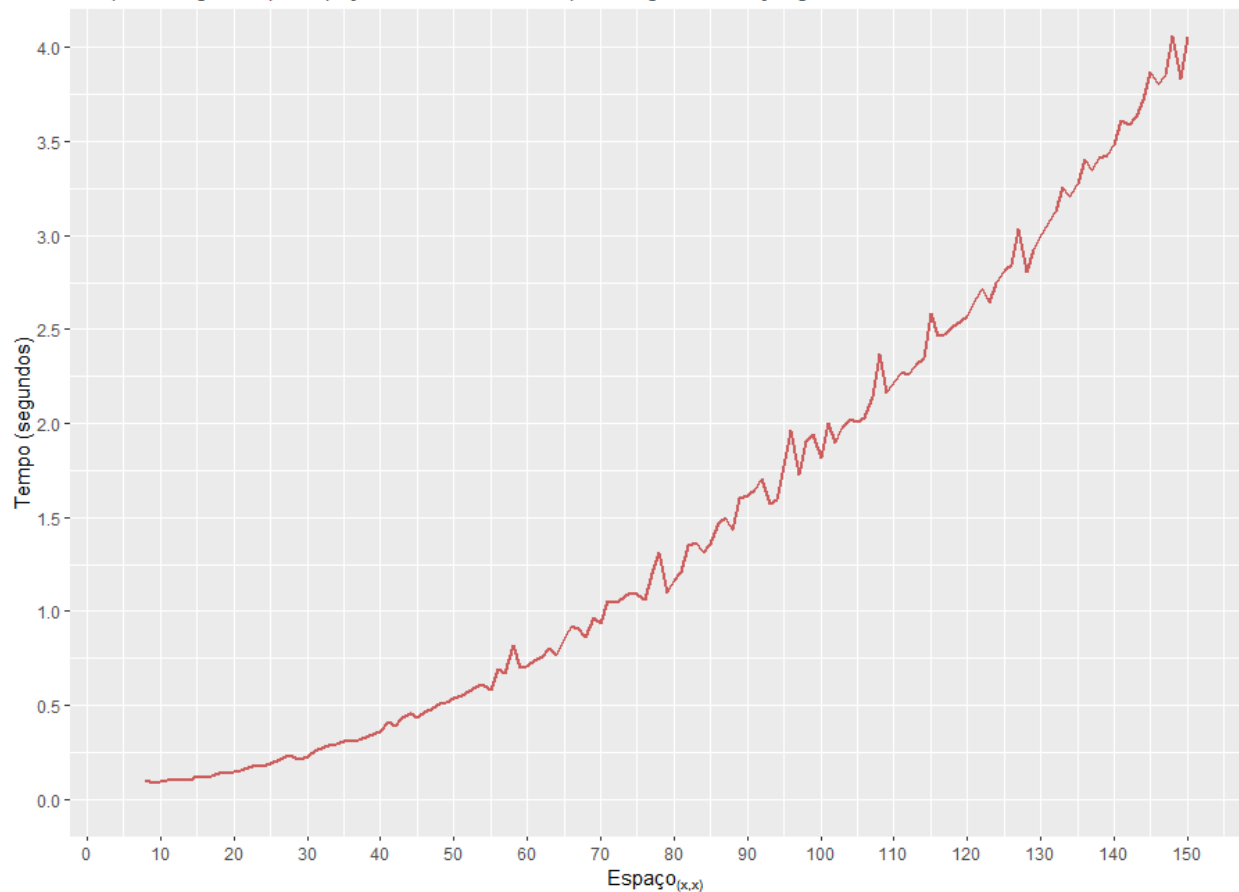
- O primeiro ponto no gráfico (  $Espaço_{(x,x)} = 8$ ,  $Nº$  de Pontos de Acesso = 20 ), corresponde a varredura do espaço de busca  $M_{(8,8)}$  que atinge a solução ótima de mínimo local com 20 Pontos de Acesso.
- Entre  $M_{(8,8)}$  e  $M_{(30,30)}$  o algoritmo passa pelos locais de 20 até 13, em  $M_{(40,40)}$  até  $M_{(53,53)}$  o algoritmo estabiliza em 13 Pa's, em  $M_{(64,64)}$  o algoritmo atinge pela primeira vez o mínimo global de 12 Pa's e entre  $M_{(65,65)}$  e  $M_{(115,115)}$  oscila entre 12 e 13 Pa's estabilizando em 12 Pa's para espaços mais densos que  $M_{(116,116)}$
- Como foram utilizados espaços de busca que começavam sempre na coordenada 1 e a maioria dos pontos de demanda não estão próximos das extremidades da área de 800 x 800, é razoável considerar que outros pontos de início para os espaços de busca talvez apresentem melhor performance.

## Custo Computacional

- Para comparar as performances do algoritmo em diferentes espaços de busca foi utilizada a função `Sys.time()` do R para verificar o tempo computacional gasto a cada interação de varredura dos espaços de busca.
- O hardware utilizado foi um processador Intel i7 com 16 GB RAM

### Custo Computacional

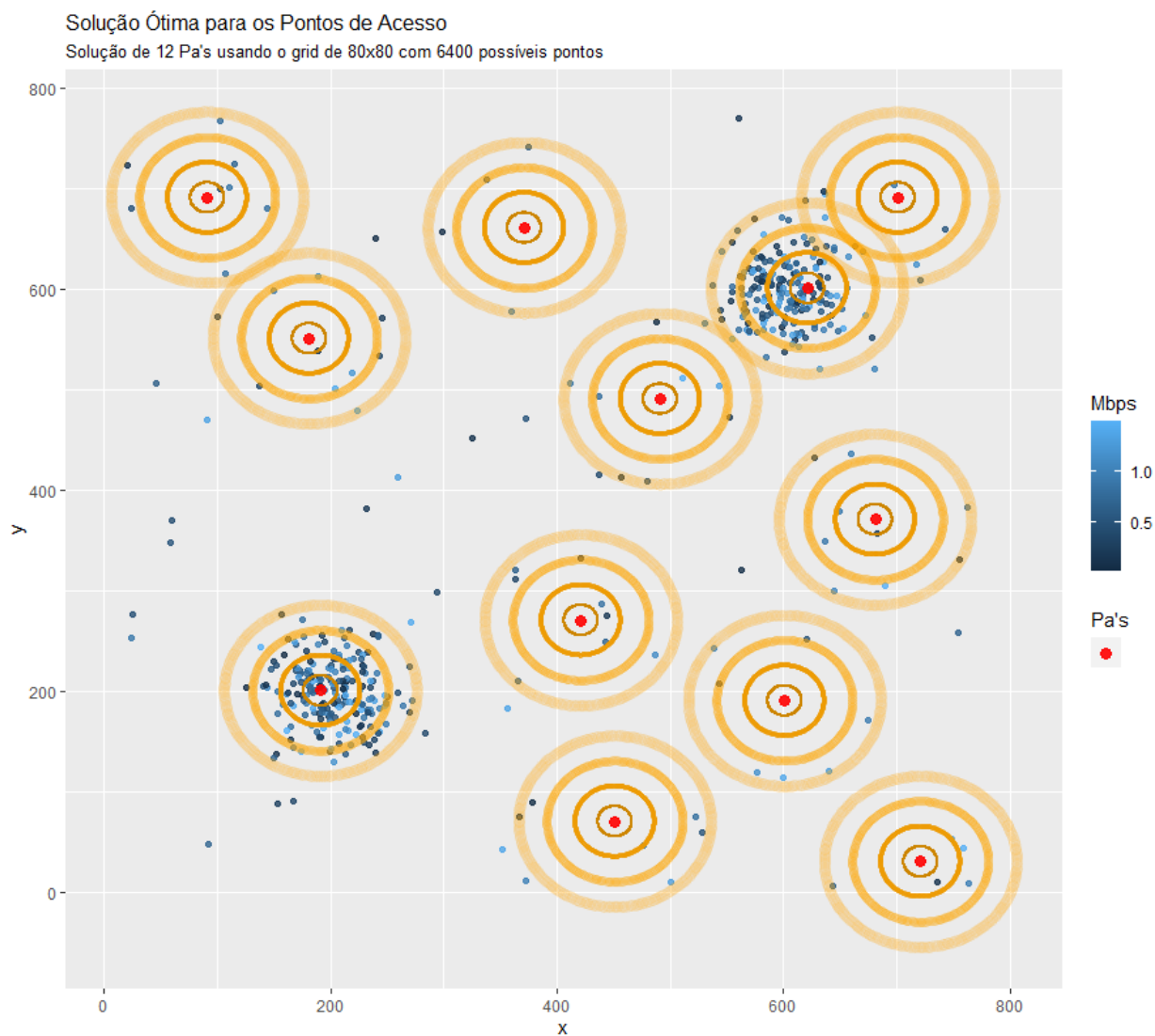
Tempo em segundos por espaço de busca necessário para atingir uma solução global ou local.



- O primeiro ponto no gráfico ( $Espaço_{(x,x)} = 8$ , Tempo (segundos) = 0,2234), corresponde a varredura do espaço de busca  $M_{(8,8)}$  que atinge a solução ótima de mínimo local com custo computacional de 0.2 segundos.
- A curva do tempo computacional apresenta leves oscilações em torno de sua tendência que provavelmente são oriundas das particularidades do funcionamento do hardware.
- Como a função de contagem de pontos foi implementada em C++, a performance apresentou resultados muito superiores se comparados aos obtidos com toda a prototipagem em R.
- O crescimento do tempo computacional é visivelmente não linear e apesar que convergir para a solução ótima global a partir de 2.5 segundos, a varredura de  $M_{(800,800)}$  com 640000 possíveis localizações dos pontos de acesso custo mais de um dia para o algoritmo e identificou o mesmo ótimo global.

## Mínimo Global

- A solução ótima considerou o espaço de busca  $M_{(80,80)}$  totalizando 6400 coordenadas de possíveis localizações para os pontos de acesso. Essa solução foi escolhida porque 80 é o ponto intermediário de um plato com 3 soluções consecutivas sobre o mínimo global, ou seja, a primeira vez que a solução global estabiliza.
- O algoritmo varreu o espaço de busca de 6400 possíveis localizações dos pontos de acesso para determinar a localização de pontos ótimos que minimizam o número de pontos de acesso e que cobrem 95 por cento da demanda sem ultrapassar o máximo de 150 Mbps para a cobertura de cada ponto de acesso.



- É possível observar que ocorre somente uma sobreposição de áreas de cobertura dos pontos de acesso no quadrante superior esquerdo do gráfico.
- Alguns pontos de acesso cobrem apenas 4 pontos de demanda e poderiam estar localizados em outras regiões cobrindo o mesmo número de pontos cobertos pela solução apresentada. Tal fato indica a existência de soluções diferentes mais igualmente eficientes apontadas pelo algoritmo de convergência.



## Tabela de Resultados

- Tabela de localização dos 12 Pa's na área de 800 x 800 com os respectivos consumos totais:

Pa	x	y	Consumo (Mbps)
1	191	201	146.5
2	621	601	146.8
3	181	551	7.4
4	681	371	6.7
5	491	491	6.2
6	421	271	4.5
7	91	691	5.0
8	701	691	5.4
9	601	191	6.6
10	451	71	3.7
11	721	31	4.6
12	371	661	2.1

- É possível observar que os dois clusters principais com um total de 146 Mbps de consumo estão muito próximos da saturação e talvez fosse adequado a colocação de mais 2 Pa's nessas regiões para garantir a integridade da entrega do sinal onde há maior concentração de pontos de demanda.
- Outra possível solução seria o deslocamento de 2 Pa's com poucos pontos de consumo para as regiões de clusters de forma que a sobreposição parcial de 2 pontos de acesso sobre os clusters principais diminua o consumo dos Pa's no limiar da saturação e também cubra os pontos vizinhos aos clusters garantindo a cobertura dos mesmos 475 pontos de demanda sem aumentar o número de Pa's.

---

## Referências

- Para desenvolver este trabalho foram utilizados os slides e videos fornecidos pelo Professor Lucas Batista do departamento de Engenharia Elétrica da UFMG.
- Devido a questões legais de proteção aos direitos autorais e regras da UFMG, os slides e videos não puderam ser compartilhados.