



RELATÓRIO - MODELAGEM MATEMÁTICA E OTIMIZAÇÃO MONO-OBJETIVO

18/04/2024

Instituição Responsável:	Universidade Federal de Minas Gerais - UFMG
Integrantes do Grupo:	Jourdan Henrique Martins Pereira Rafael Antônio Armond Maia

Sumário

1	Introdução	4
2	Modelagem Matemática	4
2.1	Variáveis de interesse	4
2.2	Parâmetros	4
2.3	Modelagem da função f_1	5
2.4	Modelagem da função f_2	5
2.5	Restrições	5
3	Algoritmo de Solução	6
3.1	O algoritmo utilizado	6
3.1.1	Ideia geral do BVNS	6
3.2	Estrutura computacional da solução	7
3.3	Estruturas de vizinhança	7
3.3.1	Estrutura de Vizinhança 1: Troca de Ativação de PA	7
3.3.2	Estrutura de Vizinhança 2: Realocação de PA's	8
3.3.3	Estrutura de Vizinhança 3: Realocação aleatória de PA's	8
3.4	Heurística construtiva para geração da solução inicial	8
4	Resultados da otimização mono-objetivo	10
4.1	Otimização da função F1	10
4.2	Otimização da função F2	19

Lista de Figuras

1	Estrutura computacional das soluções	7
2	Distribuição espacial dos clientes	9
3	Regiões de aglomeração de clientes	9
4	Pontos de acesso iniciais	10
5	Grafico de convergencia da primeira solução de F1	11
6	Conexões entre cliente e ponto de acesso da primeira solução de F1	11
7	Primeira solução de F1	12
8	Grafico de convergencia da segunda solução de F1	12
9	Conexões entre cliente e ponto de acesso da segunda solução de F1	13
10	Segunda solução de F1	13
11	Grafico de convergencia da terceira solução de F1	14
12	Conexões entre cliente e ponto de acesso da terceira solução de F1	14
13	Terceira solução de F1	15
14	Grafico de convergencia da quarta solução de F1	15
15	Conexões entre cliente e ponto de acesso da quarta solução de F1	16
16	Quarta solução de F1	16
17	Grafico de convergencia da quinta solução de F1	17
18	Conexões entre cliente e ponto de acesso da quinta solução de F1	17
19	Quinta solução de F1	18
20	Comparação das soluções encontradas para F1	18
21	Grafico de convergencia da primeira solução de F2	19
22	Conexões entre cliente e ponto de acesso da primeira solução de F2	19
23	Primeira solução de F2	20
24	Grafico de convergencia da segunda solução de F2	20
25	Conexões entre cliente e ponto de acesso da segunda solução de F2	21
26	Segunda solução de F2	21
27	Grafico de convergencia da terceira solução de F2	22
28	Conexões entre cliente e ponto de acesso da terceira solução de F2	22
29	Terceira solução de F2	23
30	Grafico de convergencia da quarta solução de F2	23
31	Conexões entre cliente e ponto de acesso da quarta solução de F2	24
32	Quarta solução de F2	24
33	Grafico de convergencia da quinta solução de F2	25
34	Conexões entre cliente e ponto de acesso da quinta solução de F2	25
35	Quinta solução de F2	26
36	Comparação das soluções encontradas para F2	26

1 Introdução

Neste trabalho, abordamos a aplicação de métodos de otimização mono-objetivo no contexto da implementação de uma rede WLAN em um centro de convenções. O objetivo principal é determinar a disposição ótima dos pontos de acesso (PAs) para maximizar a cobertura e eficiência da rede enquanto atendemos a diversas restrições operacionais e de capacidade. A importância de tal estudo reside na crescente demanda por conectividade de alta qualidade em espaços públicos amplos, onde o planejamento eficaz da infraestrutura de rede é crucial para a satisfação do usuário.

Este relatório inicia com uma formulação matemática detalhada do problema, onde definimos as funções objetivo para minimizar tanto o número de PAs necessários quanto a distância média entre os PAs e os clientes que eles servem. Adicionalmente, estabelecemos as restrições que garantem um atendimento adequado dos pontos de demanda e a capacidade máxima de cada PA. Em seguida, propomos uma adaptação de meta-heurística, estudada durante o curso, adequada para resolver este problema de otimização mono-objetivo. Detalhamos o algoritmo de solução, incluindo a modelagem de soluções candidatas, as estruturas de vizinhança para a busca local e a heurística construtiva para a geração da solução inicial.

Este trabalho não só reforça o conhecimento teórico adquirido em sala de aula, mas também proporciona uma experiência prática relevante em otimização de redes, essencial para futuras aplicações em engenharia de sistemas.

2 Modelagem Matemática

Esta seção apresenta a modelagem matemática que servirá como base para a solução prática do problema através de algoritmos de meta-heurísticas.

2.1 Variáveis de interesse

Após a leitura da especificação do problema, foi possível destacar algumas variáveis de interesse que são fundamentais na modelagem do problema.

-

$$x_{ij} : \begin{cases} 1 & \text{se o cliente } j \text{ é atendido pelo PA } i, \\ 0 & \text{caso contrário.} \end{cases} \quad (1)$$

-

$$y_i : \begin{cases} 1 & \text{se o PA } i \text{ está ativo,} \\ 0 & \text{caso contrário.} \end{cases} \quad (2)$$

2.2 Parâmetros

Segue os parâmetros levantados em sala de aula e pela análise do problema.

- n : número de clientes
- m : número de possíveis locais para instalação de PA's
- 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 euclidiana entre o PA i e o cliente j
- λ : coeficiente de exposição
- γ : fator de decaimento
- I_{ij} : exposição do cliente j ao PA i
- n_{max} : quantidade máxima de PA's disponíveis

2.3 Modelagem da função f_1

A seguinte equação representa a minimização da quantidade de PA's utilizados.

$$\min f_1 = \sum_{i=1}^m y_i \quad (3)$$

2.4 Modelagem da função f_2

Minimizar a distância entre pontos de acesso e seus respectivos clientes.

$$\min f_2 = \sum_{i=1}^m \sum_{j=1}^n x_{ij} d_{ij} \quad (4)$$

2.5 Restrições

Seguem as restrições do problema.

1. $\sum_{i=1}^m \sum_{j=1}^n x_{ij} \geq n\eta$
2. $\sum_{j=1}^n c_j x_{ij} \leq y_i q_i, \forall i \in \{1, \dots, m\}$
3. $d_{ij} x_{ij} \leq y_i r_i, \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, n\}$

4. $\sum_{x_i}^m x_{ij} y_i \geq 0,05\lambda, \forall j \in \{1, \dots, n\}$
5. $\sum_{i=1}^m x_{ij} \leq 1, \forall j \in \{1, \dots, n\}$
6. $\sum_{i=1}^m y_i \leq n_{max}$
7. $x_{ij} \in \{0, 1\}, \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, n\}$
8. $y_i \in \{0, 1\}, \forall i \in \{1, \dots, m\}$

3 Algoritmo de Solução

A seguinte seção apresenta o algoritmo que foi utilizado para encontrar a otimização mono-objetivo das funções f1 e f2 do problema em questão.

3.1 O algoritmo utilizado

O algoritmo de busca BVNS (Basic Variable Neighborhood Search) é uma meta-heurística utilizada para resolver problemas de otimização combinatória. Ele se baseia em uma busca local que explora diferentes vizinhanças em torno de uma solução corrente. Esse método alterna entre duas fases principais: shake e best improvement.

Na fase shake, são geradas soluções candidatas em diferentes vizinhanças, introduzindo alguma diversidade na busca. Em seguida, na fase best improvement, é selecionada a melhor solução dentro das vizinhanças exploradas. Esse processo é repetido até que não haja mais melhorias ou até que um critério de parada seja alcançado.

O BVNS é uma abordagem flexível e eficaz para lidar com problemas complexos, pois permite explorar várias regiões do espaço de busca de forma adaptativa, podendo ser aplicado a uma ampla gama de problemas de otimização combinatória.

3.1.1 Ideia geral do BVNS

O BVNS baseia-se na premissa de que uma mudança de vizinhança durante a busca pode levar a escapar de ótimos locais, promovendo uma exploração mais eficaz do espaço de busca. O processo pode ser descrito da seguinte forma:

1. **Inicialização:** Começa-se com uma solução inicial que pode ser gerada aleatoriamente ou por meio de alguma heurística construtiva.
2. **Shake:** Gera soluções candidatas introduzindo alguma diversidade na busca. Isso é feito explorando diferentes vizinhanças em torno da solução corrente. Cada vizinhança representa uma maneira diferente de modificar a solução atual.

3. **Best Improvement (Busca Local):** Seleciona a melhor solução dentro das vizinhanças exploradas na fase anterior.
4. **Atualização da Solução:** Se uma solução melhor for encontrada na fase Best Improvement, a solução corrente é atualizada para essa solução. Caso contrário, a solução atual permanece a mesma.
5. **Critério de Parada:** O processo de busca continua alternando entre as fases de Shake e Best Improvement até que não haja mais melhorias significativas na função objetivo ou até que um critério de parada seja atingido. Esse critério pode ser o número máximo de iterações, limite de tempo de execução ou estagnação da solução.

3.2 Estrutura computacional da solução

A soluções candidatas foram modeladas computacionalmente como uma matriz de tuplas. Cada tupla representa as coordenadas cartesianas de cada ponto de acesso (PA).

Como na figura 1 abaixo.

```
x = [(310, 125), (320, 255), (90, 365), (395, 150), (35, 95), (200, 30), (120, 175), (195, 200), (330, 60), (160, 390), (345, 115), (300, 330), (190, 320), (350, 5), (315, 270), (250, 315), (235, 215), (305, 20), (245, 30), (95, 285), (185, 100), (75, 255), (95, 35), (330, 345), (305, 330), (175, 95), (280, 165), (70, 70), (160, 235), (55, 90)]
```

Figura 1: Estrutura computacional das soluções

3.3 Estruturas de vizinhança

Estruturas de vizinhança são conjuntos de regras ou procedimentos usados em algoritmos de otimização para definir quais soluções são consideradas "próximas" ou "vizinhas" a uma solução atual. Elas permitem explorar o espaço de soluções ao modificar de forma sistemática a solução atual, buscando melhorias.

3.3.1 Estrutura de Vizinhança 1: Troca de Ativação de PA

Esta estrutura de vizinhança envolve a troca da ativação de um PA aleatoriamente escolhido. Se um PA está ativo, ele é desativado, e vice-versa. Ao ativar um novo PA, clientes próximos que estavam sendo atendidos por outros PAs podem ser realocados para reduzir a carga total ou melhorar a eficiência da rede.

1. Selecione aleatoriamente um PA.
2. Inverta o estado de ativação deste PA (de ativo para inativo ou de inativo para ativo).

3.3.2 Estrutura de Vizinhança 2: Realocação de PA's

Consiste em mover um PA aleatoriamente para alguma posição que esteja livre no grid, mas somente uma posição, ou seja, 5 metros. Pode ser em qualquer direção, sendo assim o PA pode se mover para a direita, para baixo, para a esquerda ou para cima.

1. Sorteia um número de 1 a 4. Sendo 1 mover-se para a direita, 2 mover-se para baixo, 3 mover-se para a esquerda e 4 mover-se para cima.
2. Verifica se a posição sorteada está livre, sem algum outro PA. Caso esteja move o PA para essa posição. Caso não esteja, sorteia novamente um número. Caso as quatro posições em volta do PA estejam ocupadas, nada é feito.

3.3.3 Estrutura de Vizinhança 3: Realocação aleatória de PA's

Consiste em mover um PA para uma posição completamente aleatória do grid que esteja sem um PA alocado.

1. Sortear uma posição no grid.
2. Verificar se essa posição está ocupada.
3. Caso não esteja aloca o PA para essa posição.
4. Caso esteja, sorteia uma nova posição.

3.4 Heurística construtiva para geração da solução inicial

O código possui a característica de conseguir aplicar ou não uma heurística construtiva na hora de gerar a solução inicial, através de um parâmetro que é passado para a função que gera a solução inicial. Foi implementado dessa maneira para que fosse possível comparar os resultados finais implementando uma heurística construtiva e os resultados gerando os dados iniciais de forma aleatória. Os resultados serão disponibilizados posteriormente neste relatório.

A heurística construtiva usada para geração da solução inicial se baseia na observação da distribuição de clientes no espaço. A figura 2 mostra essa distribuição. Observando a figura 2 é possível perceber que existem certos aglomerados de clientes em determinadas regiões. Essas regiões foram então delimitadas conforme mostra a figura 3.

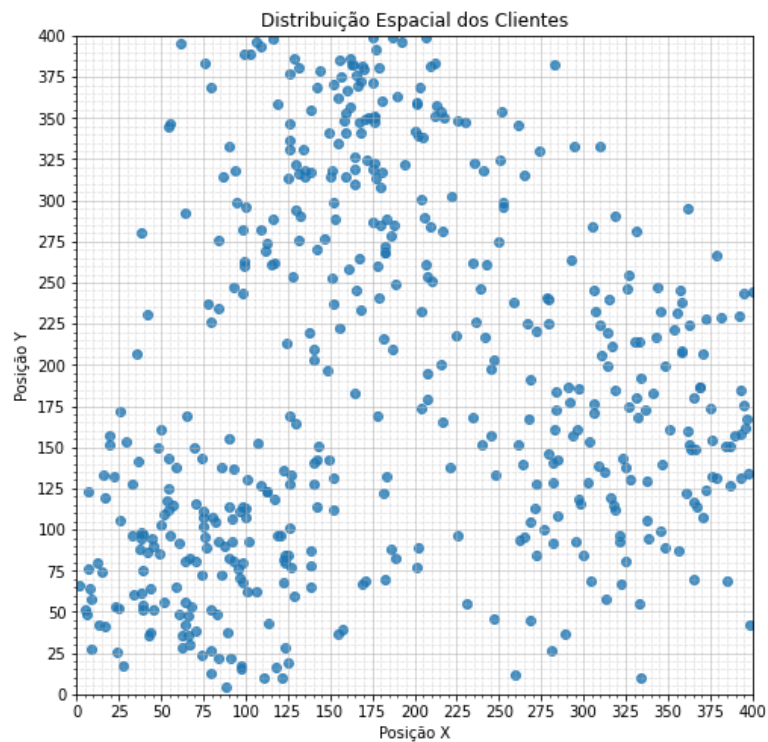


Figura 2: Distribuição espacial dos clientes

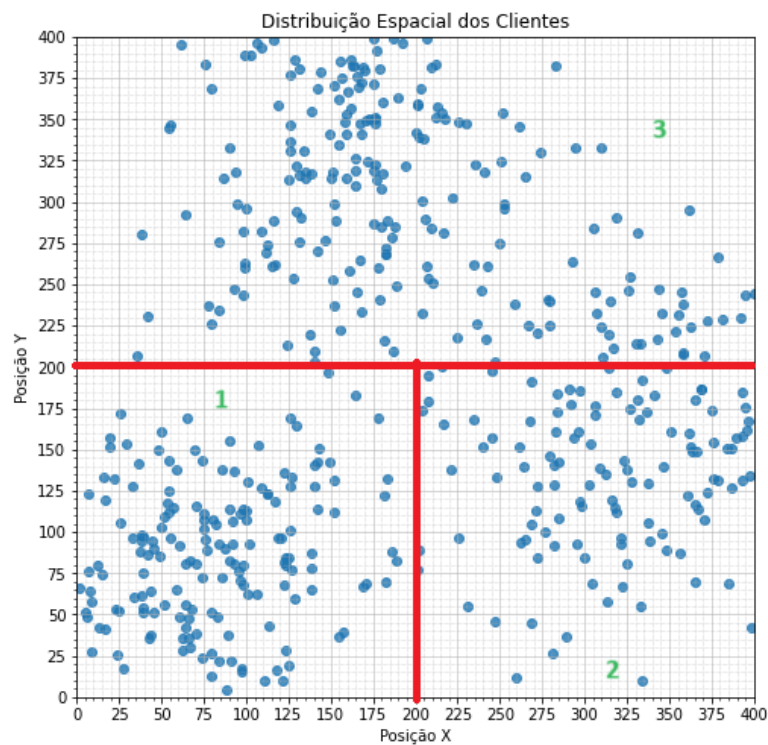


Figura 3: Regiões de aglomeração de clientes

Considerando que há 30 pontos de acesso no máximo e três regiões de aglomeração, a heurística construtiva se baseia em alocar 10 pontos de acesso em cada região. A alocação de cada ponto de acesso dentro da região se dá de forma aleatória.

4 Resultados da otimização mono-objetivo

A partir da solução inicial representada na figura 4, foi executado o BVNS para as duas funções, F1 e F2, cinco vezes cada.

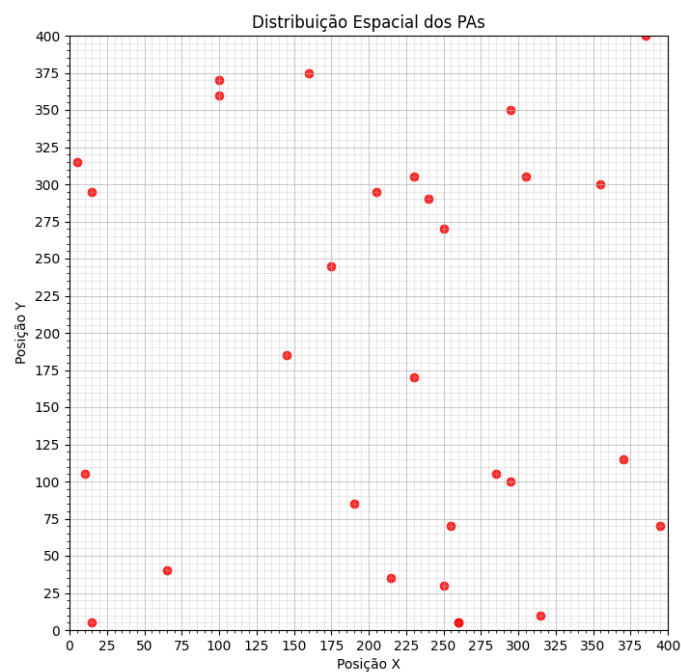


Figura 4: Pontos de acesso iniciais

4.1 Otimização da função F1

Utilizando o algoritmo implementado para rodar a função F1 temos os seguintes resultados:

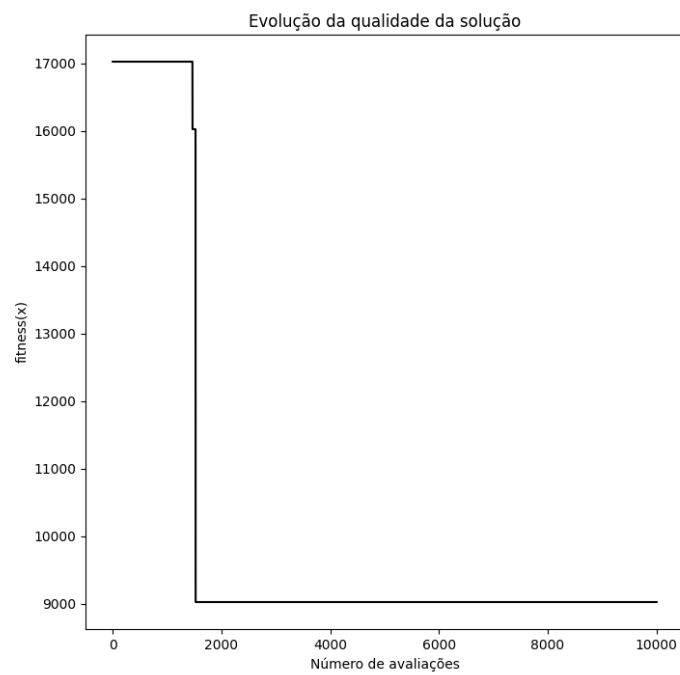


Figura 5: Grafico de convergência da primeira solução de F1

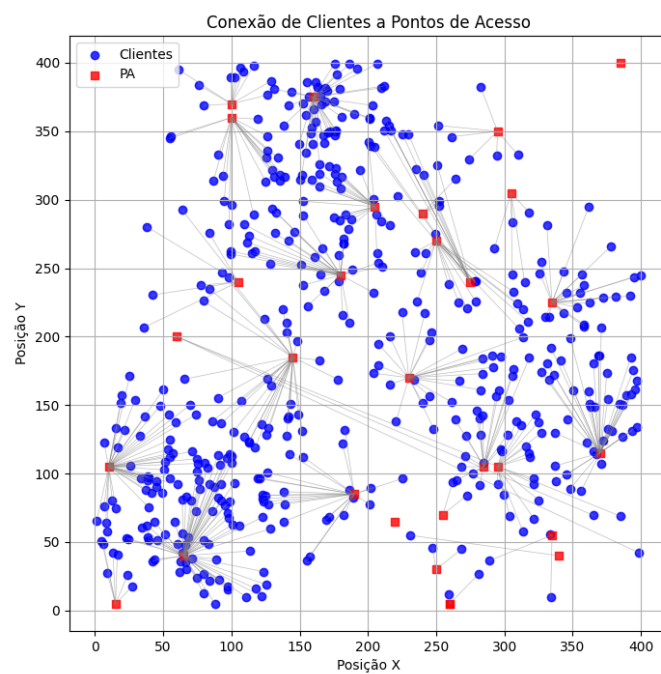


Figura 6: Conexões entre cliente e ponto de acesso da primeira solução de F1

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---

Alocação dos PAs:

x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]

fitness(x) = 17027.0

--- MELHOR SOLUÇÃO ENCONTRADA ---

Alocação dos PAs:

x = [(255, 70), (335, 225), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (220, 65), (295, 105), (160, 375), (285, 105), (240, 290), (205, 295), (60, 200), (305, 305), (275, 240), (250, 270), (335, 55), (250, 30), (100, 360), (190, 85), (105, 240), (65, 40), (260, 5), (295, 350), (340, 40), (230, 170), (385, 400), (180, 245), (15, 5)]

fitness(x) = 9027.0

Número de PAs ativos: 27

```

Figura 7: Primeira solução de F1

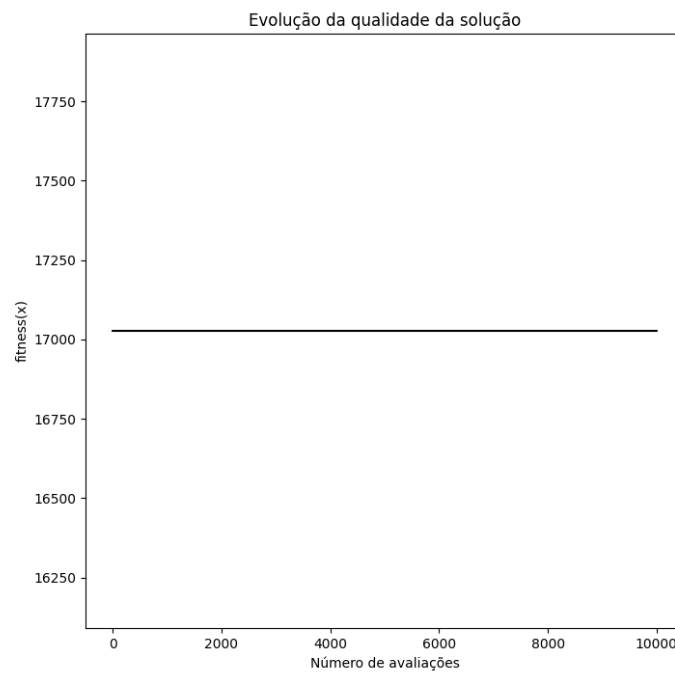


Figura 8: Grafico de convergencia da segunda solução de F1

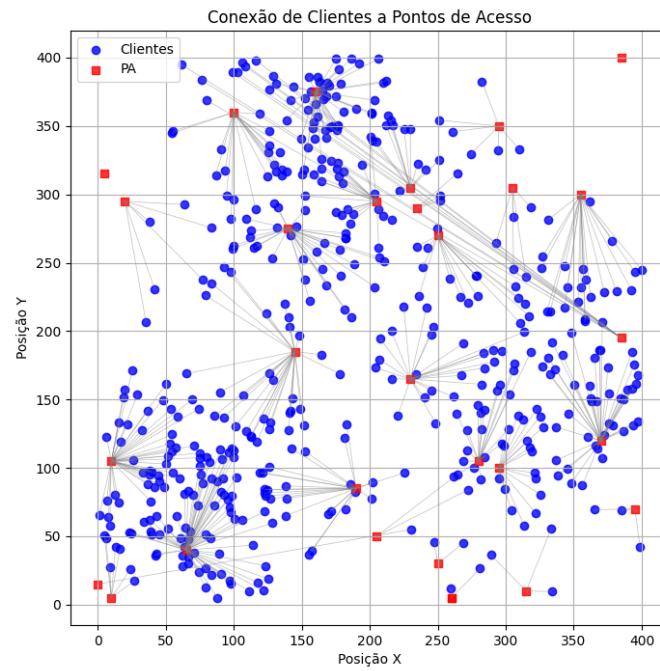


Figura 9: Conexões entre cliente e ponto de acesso da segunda solução de F1

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---

Alocação dos PAs:

x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 185), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 195), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]

fitness(x) = 17027.0

--- MELHOR SOLUÇÃO ENCONTRADA ---

Alocação dos PAs:

x = [(205, 50), (355, 300), (385, 195), (370, 120), (10, 185), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (280, 105), (235, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (20, 295), (65, 40), (260, 5), (295, 350), (0, 15), (230, 165), (385, 400), (140, 275), (10, 5)]

fitness(x) = 17027.0

Número de PAs ativos: 27

```

Figura 10: Segunda solução de F1

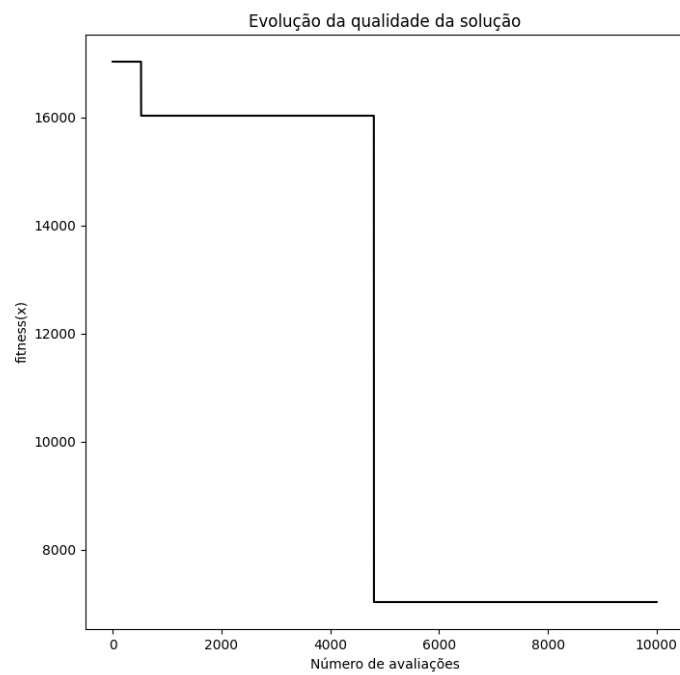


Figura 11: Grafico de convergência da terceira solução de F1

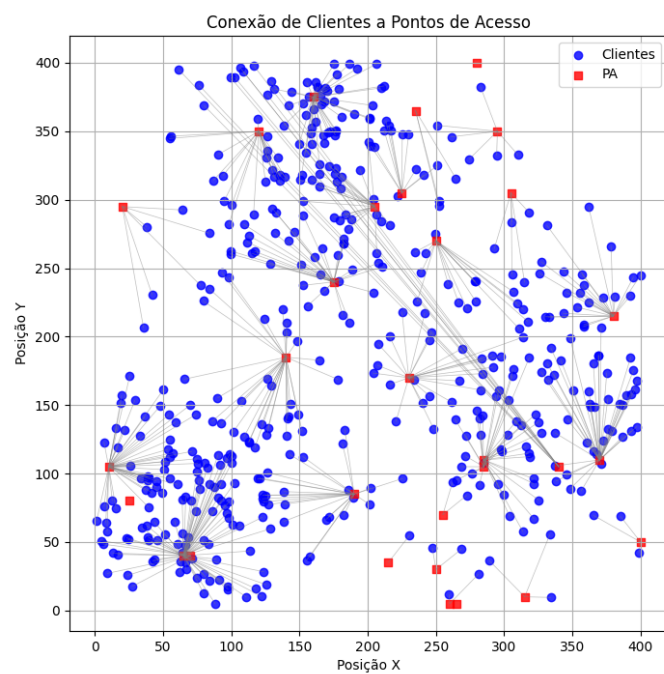


Figura 12: Conexões entre cliente e ponto de acesso da terceira solução de F1

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---

Alocação dos PAs:

x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]

fitness(x) = 17027.0

--- MELHOR SOLUÇÃO ENCONTRADA ---

Alocação dos PAs:

x = [(255, 70), (380, 215), (340, 105), (370, 110), (10, 105), (260, 5), (140, 185), (280, 400), (285, 110), (160, 375), (285, 105), (235, 365), (205, 295), (400, 50), (305, 305), (225, 305), (250, 270), (315, 10), (250, 30), (120, 350), (190, 85), (20, 295), (65, 40), (265, 5), (295, 350), (215, 35), (230, 170), (25, 80), (175, 240), (70, 40)]

fitness(x) = 7027.0

Número de PAs ativos: 27

```

Figura 13: Terceira solução de F1

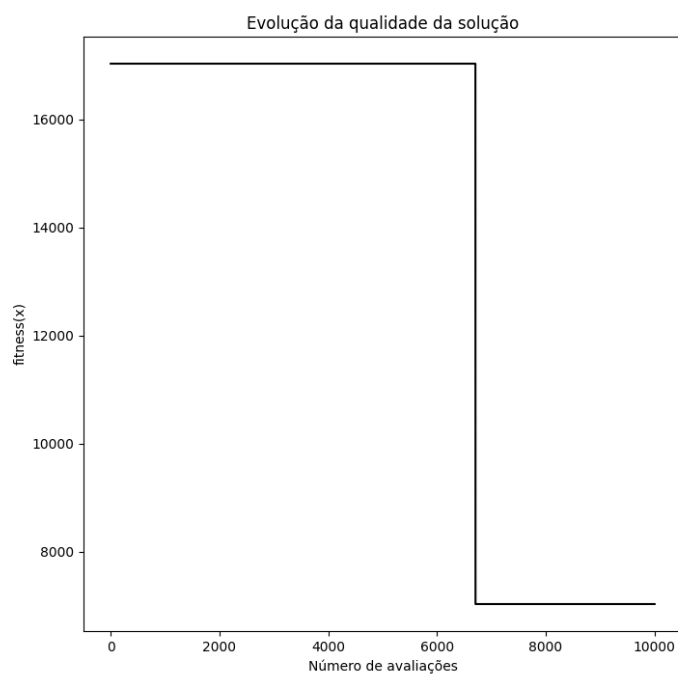


Figura 14: Grafico de convergencia da quarta solução de F1

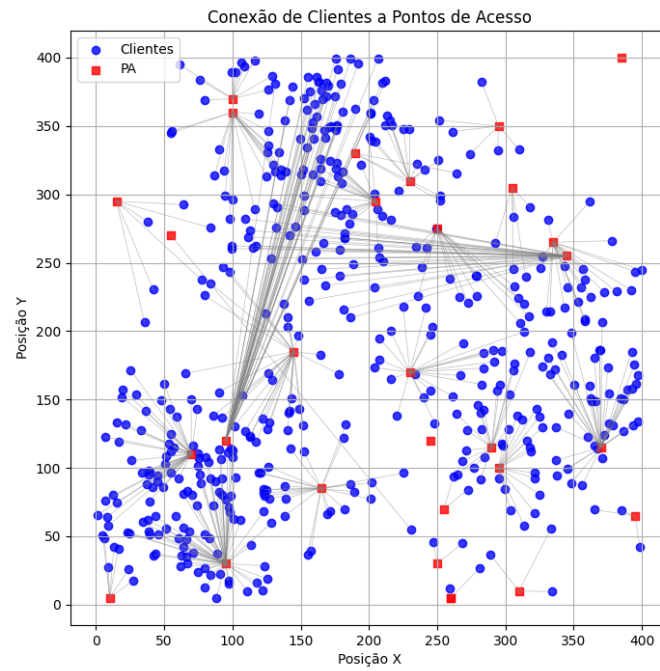


Figura 15: Conexões entre cliente e ponto de acesso da quarta solução de F1

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---
Alocação dos PAS:
x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]
fitness(x) = 17027.0

--- MELHOR SOLUÇÃO ENCONTRADA ---
Alocação dos PAS:
x = [(255, 70), (335, 265), (100, 370), (370, 115), (70, 110), (260, 5), (145, 185), (245, 120), (295, 100), (95, 120), (290, 115), (190, 330), (205, 295), (395, 65), (305, 305), (230, 310), (250, 275), (310, 10), (250, 30), (100, 360), (165, 85), (15, 295), (95, 30), (260, 5), (295, 350), (55, 270), (230, 170), (385, 400), (345, 255), (10, 5)]
fitness(x) = 7027.0

Número de PAS ativos: 27

```

Figura 16: Quarta solução de F1

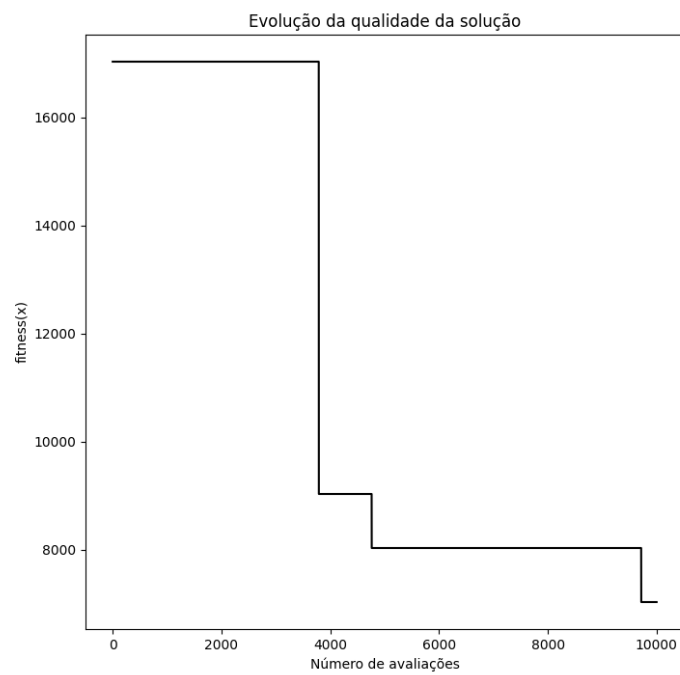


Figura 17: Grafico de convergencia da quinta solução de F1

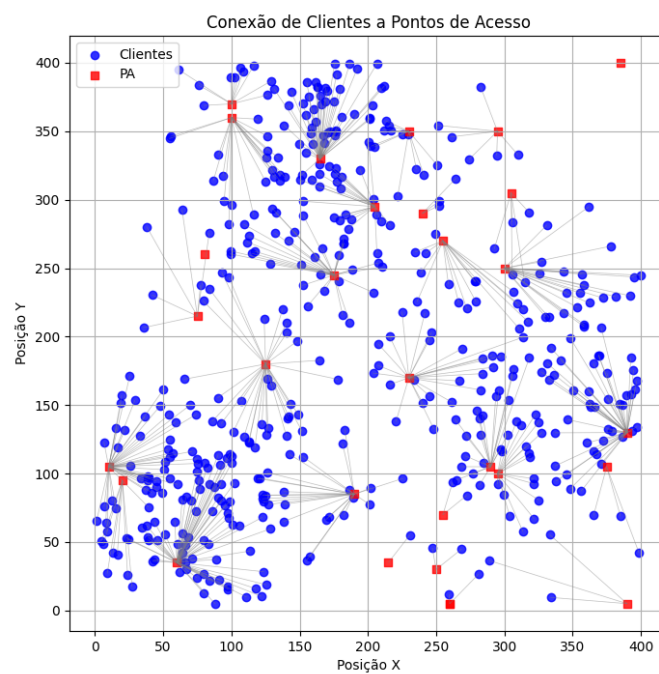


Figura 18: Conexões entre cliente e ponto de acesso da quinta solução de F1

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---
Alocação dos PAs:
x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]
fitness(x) = 17027.0

--- MELHOR SOLUÇÃO ENCONTRADA ---
Alocação dos PAs:
x = [(255, 70), (300, 250), (100, 370), (390, 130), (10, 105), (260, 5), (125, 180), (80, 260), (295, 100), (165, 330), (290, 105), (240, 290), (205, 295), (375, 105), (305, 305), (230, 350), (255, 270), (390, 5), (250, 30), (100, 360), (190, 85), (75, 215), (60, 35), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (20, 95)]
fitness(x) = 7027.0

```

Figura 19: Quinta solução de F1

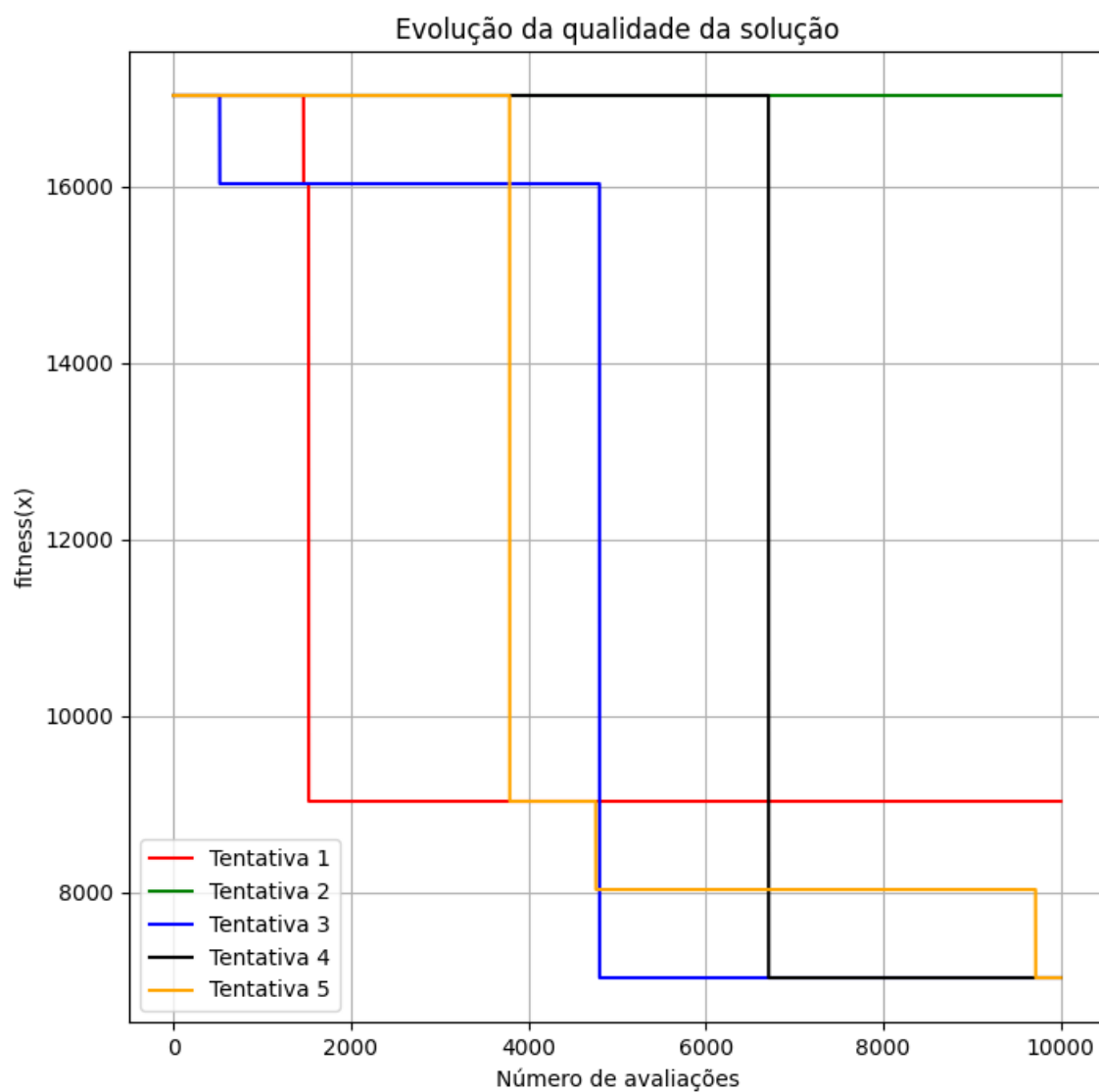


Figura 20: Comparação das soluções encontradas para F1

4.2 Otimização da função F2

Utilizando o algoritmo implementado para rodar a função F2 temos os seguintes resultados:

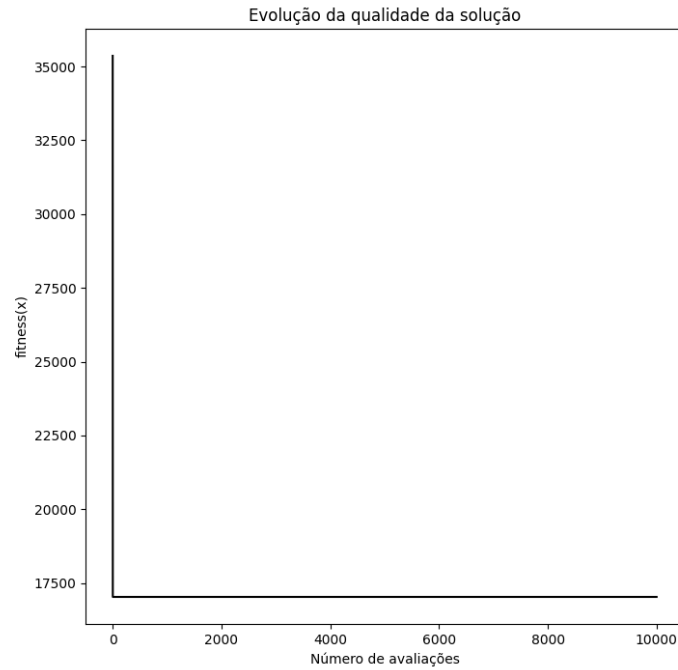


Figura 21: Grafico de convergencia da primeira solução de F2

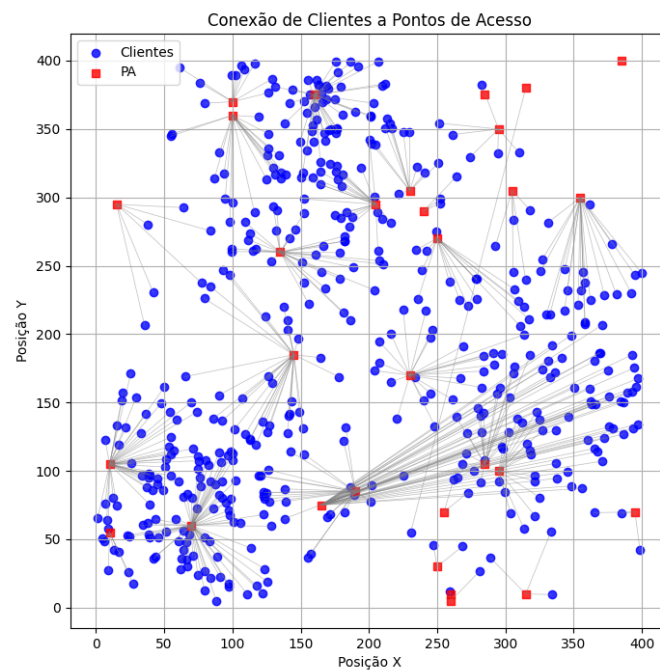


Figura 22: Conexões entre cliente e ponto de acesso da primeira solução de F2

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---

Alocação dos PAS:

x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]

fitness(x) = 35364.9

--- MELHOR SOLUÇÃO ENCONTRADA ---

Alocação dos PAS:

x = [(255, 70), (355, 300), (100, 370), (165, 75), (10, 105), (260, 5), (145, 185), (285, 375), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (70, 60), (260, 10), (295, 350), (315, 380), (230, 170), (385, 400), (135, 260), (10, 55)]

fitness(x) = 17827.0

Número de PAS ativos: 27

```

Figura 23: Primeira solução de F2

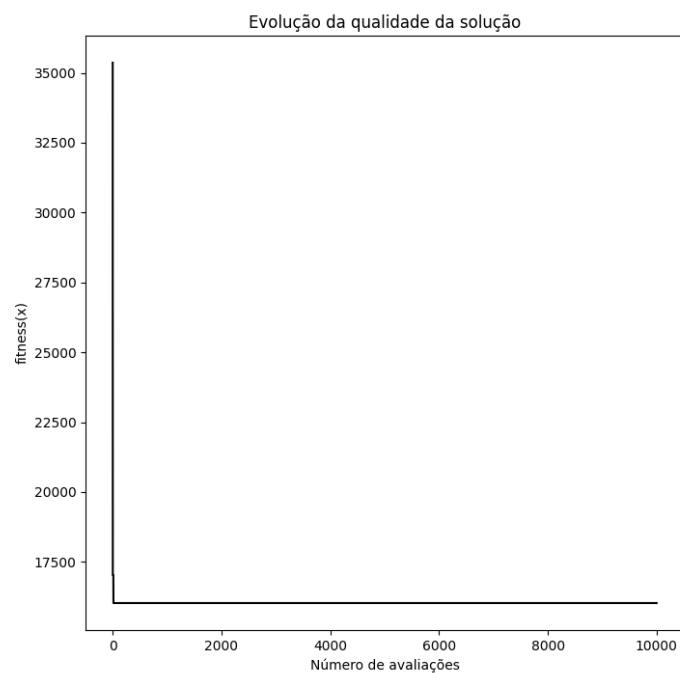


Figura 24: Grafico de convergencia da segunda solução de F2

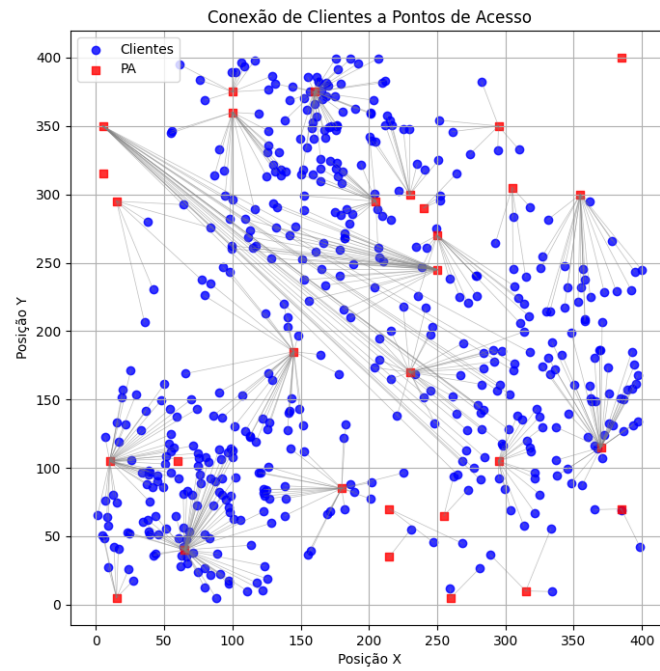


Figura 25: Conexões entre cliente e ponto de acesso da segunda solução de F2

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---

Alocação dos PAs:

x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]

fitness(x) = 35364.9

--- MELHOR SOLUÇÃO ENCONTRADA ---

Alocação dos PAs:

x = [(255, 65), (355, 300), (100, 375), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 105), (160, 375), (5, 350), (240, 290), (205, 295), (385, 70), (305, 305), (230, 300), (250, 270), (315, 10), (215, 70), (100, 360), (180, 85), (15, 295), (65, 40), (60, 105), (295, 350), (215, 35), (230, 170), (385, 400), (250, 245), (15, 5)]

fitness(x) = 16027.0

Número de PAs ativos: 27

```

Figura 26: Segunda solução de F2

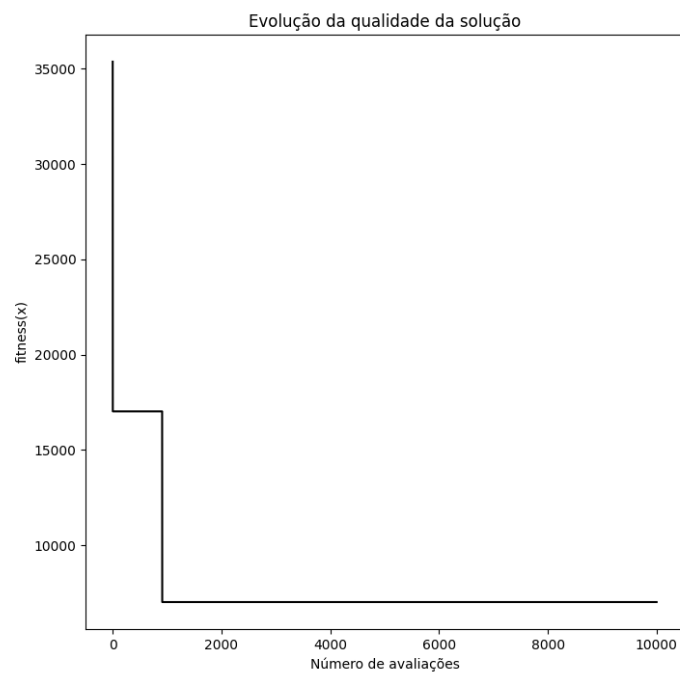


Figura 27: Grafico de convergencia da terceira solução de F2

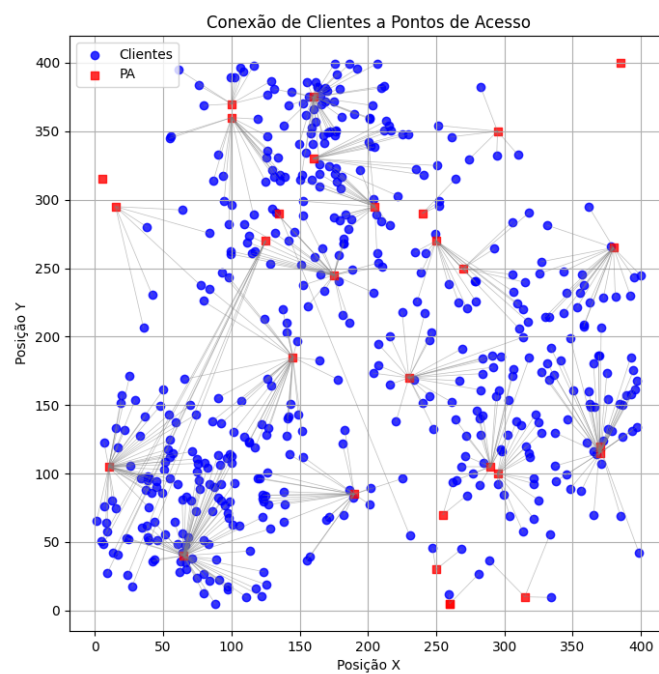


Figura 28: Conexões entre cliente e ponto de acesso da terceira solução de F2

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---

Alocação dos PAS:

x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (100, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]

fitness(x) = 35364.9

--- MELHOR SOLUÇÃO ENCONTRADA ---

Alocação dos PAS:

x = [(255, 70), (380, 265), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (290, 105), (240, 290), (205, 295), (370, 120), (270, 250), (160, 330), (250, 270), (315, 10), (250, 30), (100, 360), (100, 85), (15, 295), (65, 40), (260, 5), (295, 350), (135, 290), (230, 170), (385, 400), (175, 245), (125, 270)]

fitness(x) = 7027.0

Número de PAS ativos: 27

```

Figura 29: Terceira solução de F2

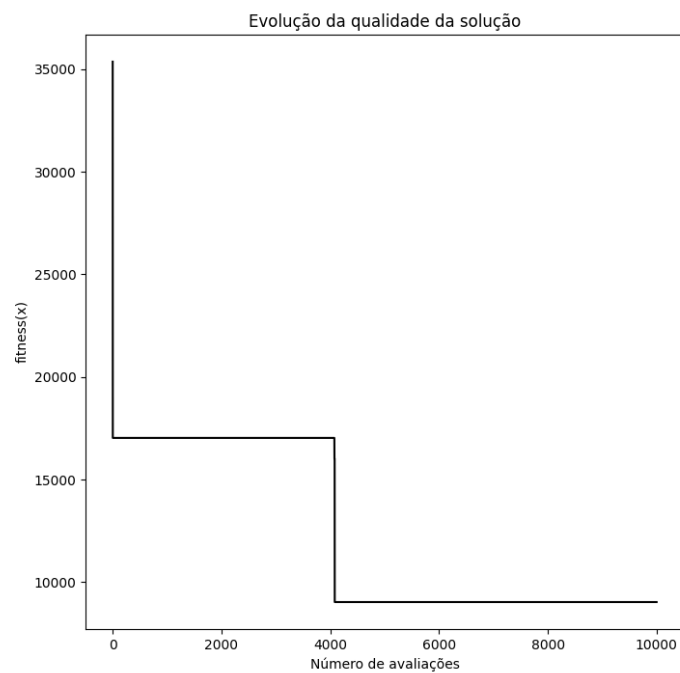


Figura 30: Grafico de convergencia da quarta solução de F2

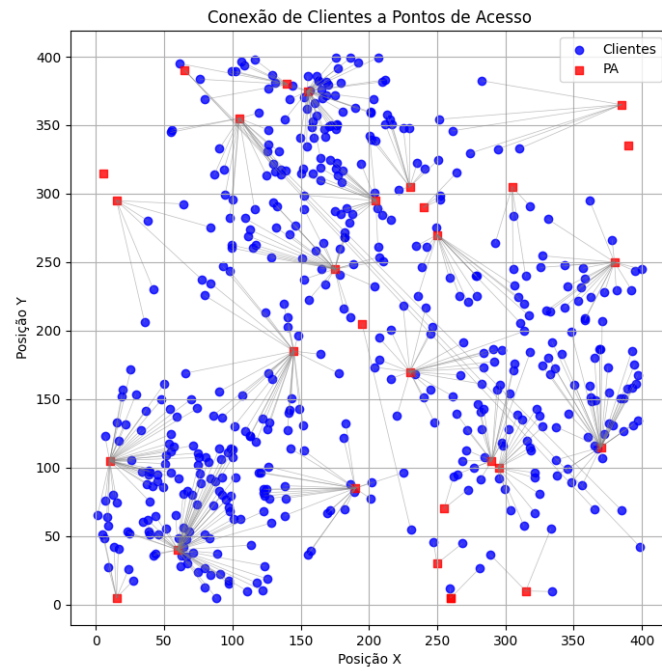


Figura 31: Conexões entre cliente e ponto de acesso da quarta solução de F2

```
Alocação dos PAs:
x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]
fitness(x) = 35364.9

--- MELHOR SOLUÇÃO ENCONTRADA ---
Alocação dos PAs:
x = [(255, 70), (380, 250), (140, 380), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (155, 375), (290, 105), (240, 290), (205, 295), (65, 390), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (105, 355), (190, 85), (15, 295), (60, 40), (260, 5), (385, 365), (195, 205), (230, 170), (390, 335), (175, 245), (15, 5)]
fitness(x) = 9027.0
Número de PAs ativos: 27
```

Figura 32: Quarta solução de F2

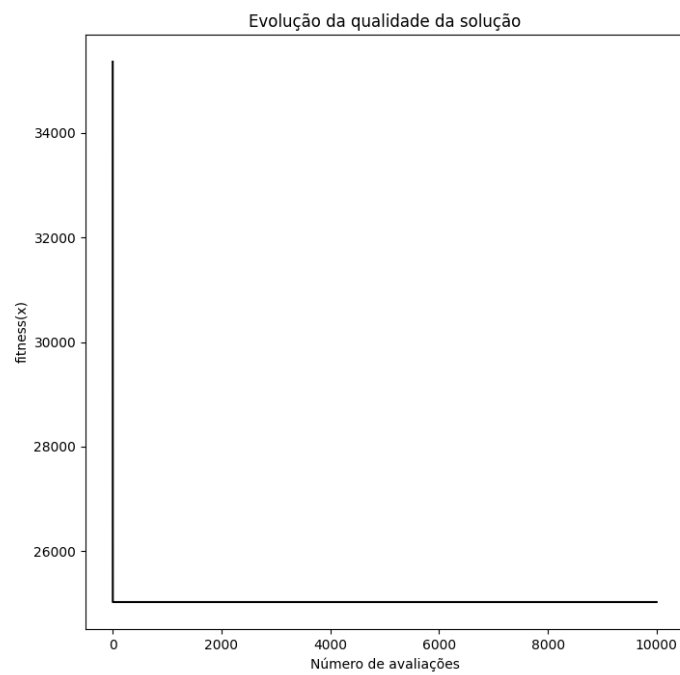


Figura 33: Grafico de convergencia da quinta solução de F2

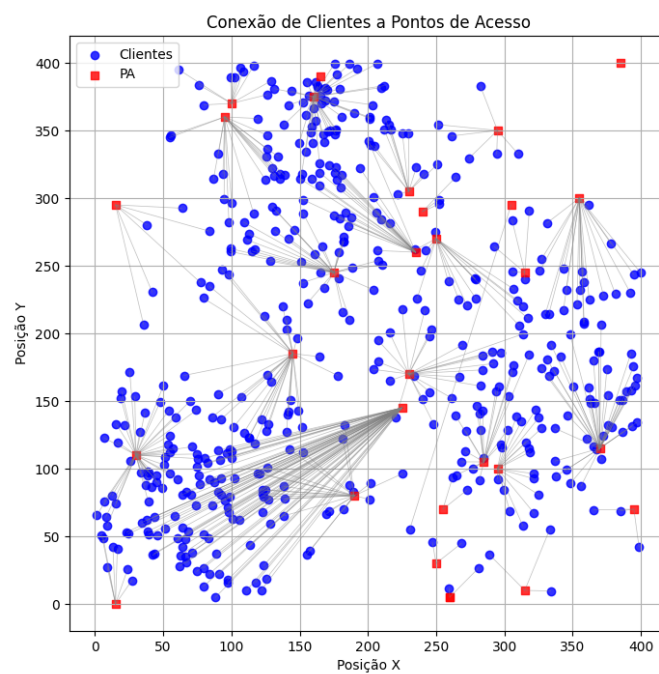


Figura 34: Conexões entre cliente e ponto de acesso da quinta solução de F2

```

--- SOLUÇÃO INICIAL CONSTRUÍDA ---
Alocação dos PAS:
x = [(255, 70), (355, 300), (100, 370), (370, 115), (10, 105), (260, 5), (145, 185), (5, 315), (295, 100), (160, 375), (285, 105), (240, 290), (205, 295), (395, 70), (305, 305), (230, 305), (250, 270), (315, 10), (250, 30), (100, 360), (190, 85), (15, 295), (65, 40), (260, 5), (295, 350), (215, 35), (230, 170), (385, 400), (175, 245), (15, 5)]
fitness(x) = 35364.9

--- MELHOR SOLUÇÃO ENCONTRADA ---
Alocação dos PAS:
x = [(255, 70), (355, 300), (100, 370), (370, 115), (30, 110), (260, 5), (145, 185), (165, 300), (295, 100), (160, 375), (285, 105), (240, 290), (235, 260), (395, 70), (315, 245), (230, 305), (250, 270), (315, 10), (250, 30), (95, 360), (190, 80), (15, 295), (225, 145), (260, 5), (295, 350), (305, 295), (230, 170), (385, 400), (175, 245), (15, 0)]
fitness(x) = 25027.0

Número de PAS ativos: 27

```

Figura 35: Quinta solução de F2

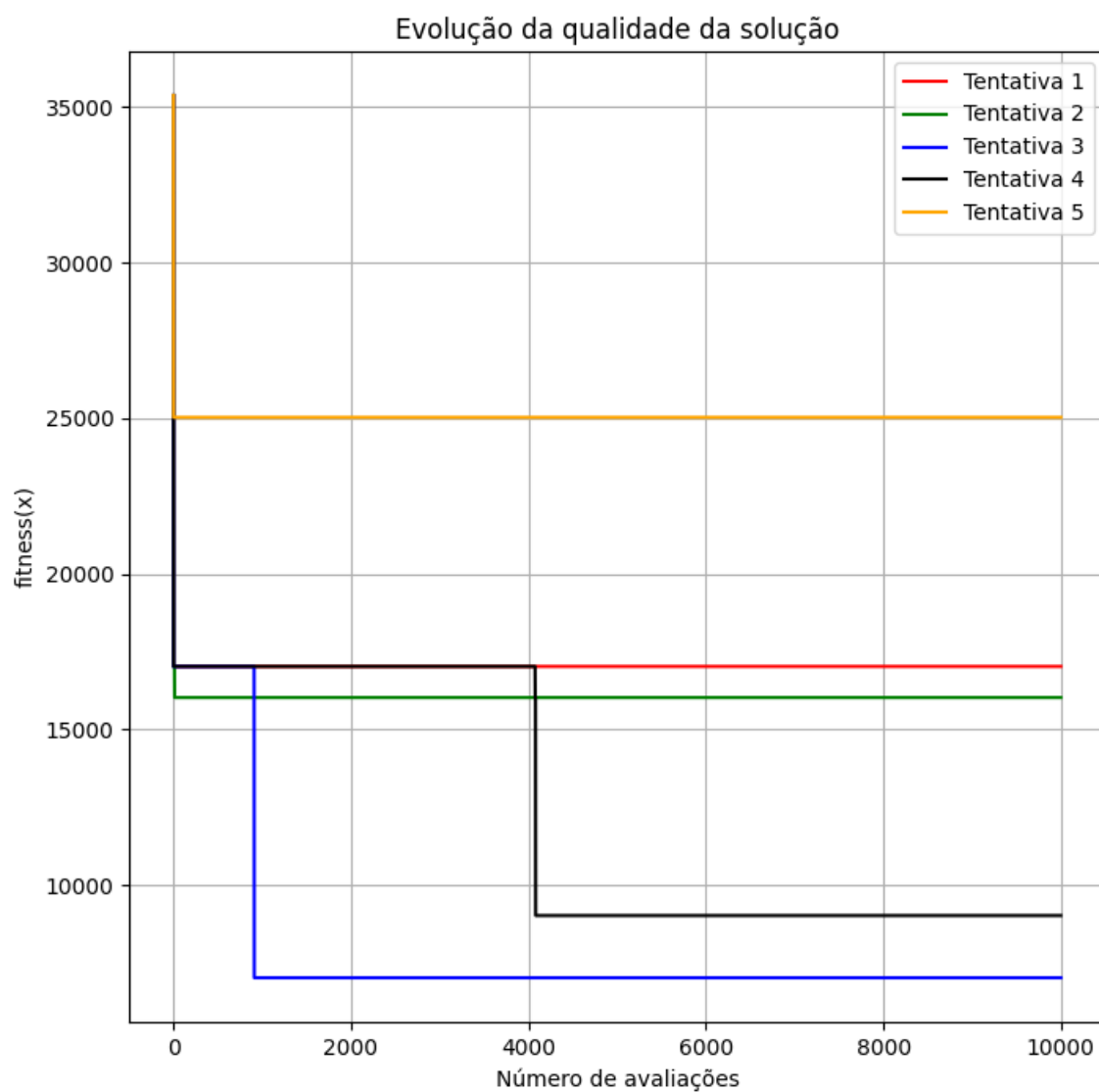


Figura 36: Comparação das soluções encontradas para F2