



MODELAGEM MATEMÁTICA E OTIMIZAÇÃO MULTI-OBJETIVO

14/07/2024

Título: Teoria da Decisão - Trabalho Computacional 2

Instituição Responsável: Universidade Federal de Minas Gerais (UFMG)

Nome do(a) autor(a):
Bernardo Araújo Ribeiro
Marcus Vinicius Ferreira Vilela
Maria Luiza Coelho Rodrigues

Sumário

1	Introdução	3
2	Modelagem Matemática	3
2.1	Variáveis de interesse	3
2.2	Parâmetros	4
2.3	Modelagem da função f_1	4
2.4	Modelagem da função f_2	4
2.5	Restrições	5
3	Algoritmo de Solução	5
3.1	Solução Inicial	7
3.2	Algoritmo de busca	7
3.3	Estruturas de vizinhança	8
4	Resultados da otimização multi-objetivo	9
4.1	e-Restrito	9
4.2	Soma Ponderada	9

1 Introdução

Este trabalho computacional abordada a aplicação de métodos de otimização multi-objetivo no contexto da implementação de uma rede WLAN do tipo N 2D para atendimento de um centro de convenções com 400×400m. Considerando a necessidade de conexões de alta qualidade em espaços públicos, o objetivo central é deste relatório é definir a disposição ótima dos pontos de acesso (PAs) para maximizar a cobertura e eficiência da rede respeitando as restrições do problema de maximização, por meio de um bom planejamento de infraestrutura de rede.

O relatório está dividido em três partes principais:

- **Modelagem Matemática** - definição das funções objetivo para minimizar o número de PAs, a distância média entre os PAs e os clientes que eles servem, atendendo as restrições operacionais e de capacidade.
- **Algoritmo de Solução** - aplicação de uma meta-heurística apropriada, discutida em sala de aula durante o curso de Teoria da Decisão, com ajustes nos parâmetros necessários para solucionar a otimização Multi-objetivo da rede WLAN. Exploração da modelagem de soluções candidatas, estruturas de vizinhança e heurística construtiva.
- **Resultados da Otimização Multi-Objetivo** - apresentação dos resultados alcançados por meio da otimização Multi-objetivo, destacando as descobertas e conclusões obtidas durante o processo de análise e modelagem.

Por fim, é importante ressaltar que este trabalho consolida o conhecimento teórico adquirido na disciplina de Teoria da Decisão, por meio da aplicação prática no contexto da implementação de uma rede WLAN.

2 Modelagem Matemática

O processo de modelagem matemática do problema proposto é importante para a construção do algoritmo de meta-heurística, uma vez que viabiliza a formalização da implementação de uma rede WLAN, definindo critérios de avaliação, identificando estratégias de busca e colaborando com a validação dos resultados.

2.1 Variáveis de interesse

A seguir estão definidas as variáveis de interesse para a 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

A seguir estão os parâmetros para a modelagem matemática:

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

2.3 Modelagem da função f_1

A equação a seguir representa a minimização da quantidade de pontos de acesso (PA) utilizados:

$$\min f_1 \sum_i^m y_i \quad (3)$$

2.4 Modelagem da função f_2

A equação a seguir representa a minimização da distância entre os 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

As equações a seguir representam as restrições do problema:

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} \geq n\eta \quad (5)$$

$$\sum_{j=1}^n c_j x_{ij} \leq y_i q_i, \forall i \in \{1, \dots, m\} \quad (6)$$

$$d_{ij} x_{ij} \leq y_i r_i, \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, n\} \quad (7)$$

$$\sum_{x_i}^m x_{ij} y_i \geq 0,05\lambda, \forall j \in \{1, \dots, n\} \quad (8)$$

$$\sum_{i=1}^m x_{ij} \leq 1, \forall j \in \{1, \dots, n\} \quad (9)$$

$$\sum_{i=1}^m y_i \leq n_{max} \quad (10)$$

$$x_{ij} \in \{0, 1\}, \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, n\} \quad (11)$$

$$y_i \in \{0, 1\}, \forall i \in \{1, \dots, m\} \quad (12)$$

3 Algoritmo de Solução

Otimização Mono Objetivo

Apesar do foco dessa segunda entrega ser a minimização multi-objetivo, é importante explicar inicialmente como foi modelada a minimização mono objetivo, uma para cada problema a ser analisado.

Dessa forma no algoritmo de solução foi criado duas funções objetivos que foram nomeadas **fobj1** e **fobj2**.

Função Objetivo 1

A modelagem da função objetivo *fobj1* visa minimizar o número de pontos de acesso (PAs) ativos necessários para atender a todos os clientes, ao mesmo tempo em que penaliza a existência de clientes não atendidos ou fora da distância máxima permitida.

Para realizar essa modelagem, a função itera sobre cada cliente e verifica se está atribuído a um ponto de acesso. Caso um cliente não esteja atribuído a nenhum PA (ou seja, $pa_index == -1$), ele é contado como não atendido (*unserved_clients* é incrementado). Isso é importante porque o objetivo é atender o maior número possível de clientes para minimizar a necessidade de pontos de acesso.

Além disso, para cada cliente atribuído a um PA, a função calcula a distância entre o cliente e o PA correspondente. Se essa distância excede um limite predefinido, um total de penalidade é acumulado. Isso incentiva a atribuição dos clientes aos PAs mais próximos, minimizando assim a necessidade de instalar novos PAs.

A penalidade acumulada é baseada no número de clientes não atendidos que excedem o limite permitido (*allowed_unserved*). Se esse número ultrapassa o limite, uma penalidade é aplicada multiplicando o número de clientes não atendidos pelo valor de penalidade (100 neste caso).

Finalmente, a função fitness da solução é determinada pela soma do número de PAs ativos ($len(active_pas)$) e da penalidade total acumulada (*total_penalty*). Esse valor de fitness é utilizado para avaliar a solução dentro do contexto da otimização, onde o objetivo é minimizar o número de PAs ativos.

Função Objetivo 2

A modelagem da função objetivo *fobj2* visa minimizar a distância total entre os clientes atendidos e seus pontos de acesso correspondentes, enquanto penaliza a existência de clientes não atendidos ou fora da distância máxima permitida.

Nesta função objetivo, a abordagem é semelhante à função *fobj1*, mas com o foco na minimização da distância total entre clientes e seus pontos de acesso atribuídos. Para cada cliente atribuído a um ponto de acesso válido (ou seja, $pa_index \neq -1$), a função calcula a distância euclidiana entre o cliente e o ponto de acesso correspondente.

A distância entre o cliente e seu ponto de acesso é somada à *total_distance*, que representa a soma acumulada das distâncias de todos os clientes atendidos. Isso incentiva a atribuição dos clientes aos pontos de acesso mais próximos, o que, por sua vez, ajuda a minimizar a distância total.

A função fitness da solução é determinada pela soma da *total_distance* e *total_penalty*. Portanto, para otimizar essa função objetivo, o algoritmo de otimização busca uma solução que minimize tanto a distância total entre clientes e PAs quanto o número de clientes não atendidos.

Essa abordagem visa encontrar uma configuração de PAs que minimize a distância média de comunicação entre clientes e PAs atribuídos, enquanto atendendo ao máximo de clientes possível e respeitando as restrições do problema.

3.1 Solução Inicial

Para a solução inicial, foi utilizado a estratégia de clustering para identificar as regiões com mais concentração de clientes e alocar um PA nesse local. Para isso, é necessário definir a quantidade de regiões que se deseja identificar inicialmente. Para a aplicação desse conceito foi utilizado o K-means da biblioteca SKLearn.

K-means

O K-Means é um algoritmo de agrupamento não supervisionado implementado na biblioteca Scikit-Learn, usada em Python. Este método particiona um conjunto de dados em k grupos ou clusters, onde k é um número pré-definido de clusters. O objetivo do K-Means é minimizar a variabilidade dentro de cada cluster e maximizar a variabilidade entre os clusters. Isso é feito iterativamente: primeiro, são escolhidos aleatoriamente k centroides iniciais. Em seguida, cada ponto de dados é atribuído ao centróide mais próximo, formando os clusters. Após isso, os centroides são recalculados com base nas médias dos pontos atribuídos a cada cluster, e o processo se repete até que os centroides não mudem significativamente ou um número máximo de iterações seja alcançado.

Para usar o K-Means no Scikit-Learn, é necessário importar o módulo KMeans e definir o número de clusters desejado. Após ajustar o modelo aos dados com o método `fit`, podemos obter as previsões de cluster para novos dados utilizando o método `predict`. Além disso, o Scikit-Learn fornece métricas para avaliar a qualidade dos clusters, como o valor inercial, que mede a soma das distâncias quadráticas dos pontos de dados aos seus centroides respectivos. O K-Means é amplamente utilizado em diversas aplicações, incluindo segmentação de clientes, compressão de imagem e agrupamento de documentos, devido à sua simplicidade e eficiência.

3.2 Algoritmo de busca

O algoritmo BVNS (Basic Variable Neighborhood Search) é uma meta-heurística poderosa para resolver problemas complexos de otimização combinatória. Ele se destaca por sua capacidade de explorar eficientemente diferentes regiões do espaço de busca, adaptando-se dinamicamente às características do problema.

A ideia central do BVNS é realizar uma busca local que alterna entre duas fases principais: *shake* (agitamento) e *best improvement* (melhor melhoria). Na fase de *shake*, o algoritmo gera soluções candidatas explorando várias vizinhanças em torno da solução atual, introduzindo diversidade na busca. Em seguida, na fase de *best improvement*, ele seleciona a melhor solução encontrada entre as vizinhanças exploradas.

O processo começa com uma solução inicial, que pode ser gerada aleatoriamente ou por meio de uma heurística construtiva. Durante a fase de shake, diferentes vizinhanças são exploradas, cada uma representando uma modificação distinta na solução atual. Em seguida, na fase de best improvement, a melhor solução dentro das vizinhanças é selecionada e se torna a nova solução corrente, se for melhor do que a solução atual.

O algoritmo 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, como um número máximo de iterações, limite de tempo ou estagnação da solução.

O BVNS é altamente flexível e pode ser aplicado a uma ampla variedade de problemas de otimização combinatória, destacando-se por sua eficácia em explorar o espaço de busca de forma adaptativa e eficiente. Esta abordagem permite escapar de ótimos locais e encontrar soluções de alta qualidade em problemas desafiadores.

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.

Realocação Dentro de uma Região de um PA

Consiste em mover um PA para uma posição delimitada no grid, desde que a posição esteja livre. Esse método permite explorar as regiões do espaço de busca.

Procedimento:

- Seleção aleatória de um PA.
- Geração de uma nova posição no grid.
- Verificação da disponibilidade da nova posição e realocação do PA, se possível.

As estruturas de vizinhança utilizadas foram realocações de um PA, dentro de uma região de 10 metros e depois de 50 metros.

Ativação ou Desativação de um PA

Consiste em desativar ou ativar um PA para tentar escapar de mínimos locais.

Procedimento:

- Sortear se vai ser uma ativação ou desativação, considerando uma chance de 0,5 para cada uma.
- Buscar um PA que se adeque as condições para ser ativado ou desativado.
- Realizar a alteração.

Esse procedimento foi escolhido com a última estrutura de vizinhança.

4 Resultados da otimização multi-objetivo

Os resultados obtidos na segunda entrega do projeto foram extremamente promissores, destacando o sucesso da abordagem em resolver os problemas de minimização.

Para obter os resultados, foram utilizados os métodos e-restrito e soma ponderada.

4.1 e-Restrito

Essa abordagem transforma o problema multiobjetivo em um problema mono-objetivo com restrições adicionais, onde uma função é minimizada e as demais viram restrições do problema.

Para aplicação, a minimização do número de PA's foi considerada uma restrição e foi aplicada variando os valores de 30 até 10. Após isso, solucionamos como um problema mono objetivo.

As cinco execuções para o método e-Restrito:

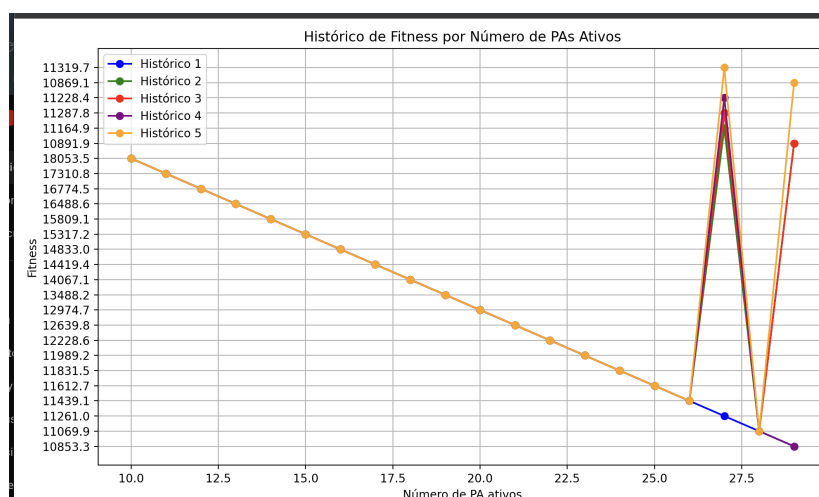


Figura 1: Fronteira pareto epsilon restrito

4.2 Soma Ponderada

O método da soma ponderada é uma técnica utilizada para transformar um problema de otimização multiobjetivo em um problema de otimização mono-objetivo. Isso é feito ao atribuir pesos a

cada função objetivo individual, refletindo a importância relativa de cada objetivo, e somando essas funções ponderadas em uma única função objetivo. No contexto do nosso problema de alocação de Pontos de Acesso (PAs), temos dois objetivos conflitantes: minimizar o número de PAs ativos e minimizar a distância total entre clientes e PAs. Cada função objetivo é calculada separadamente, normalizada para uma escala comum, e então combinada em uma função objetiva única utilizando os pesos definidos.

A resposta do modelo para o método Soma Ponderada:

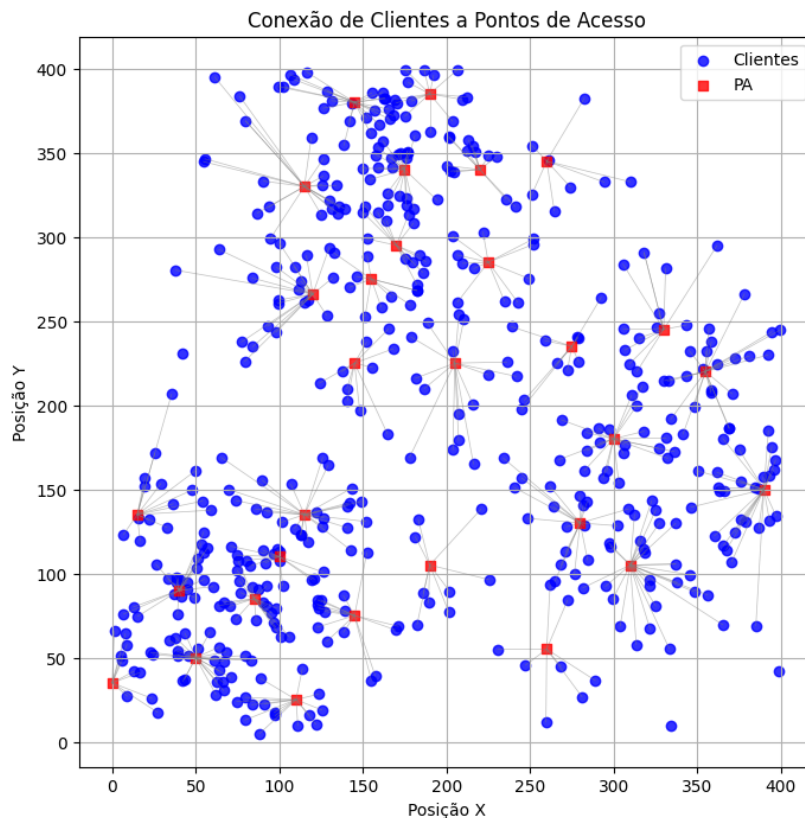


Figura 2: Relação clientes e PA's em uma das solucoes da fronteira

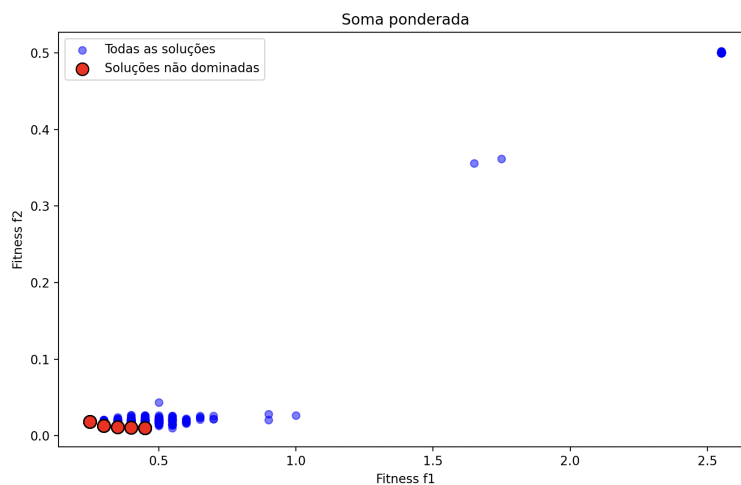


Figura 3: Solucoes nao dominadas