Algoritmos Gulosos

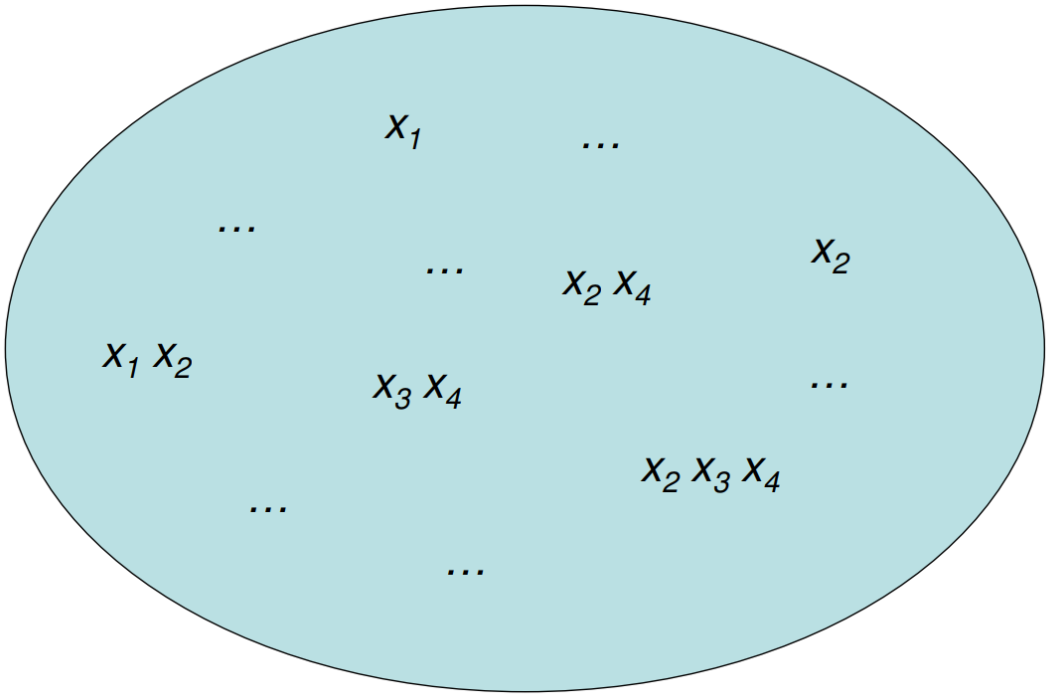
Prof. Fábio Dias

22 de janeiro de 2025

Sumário

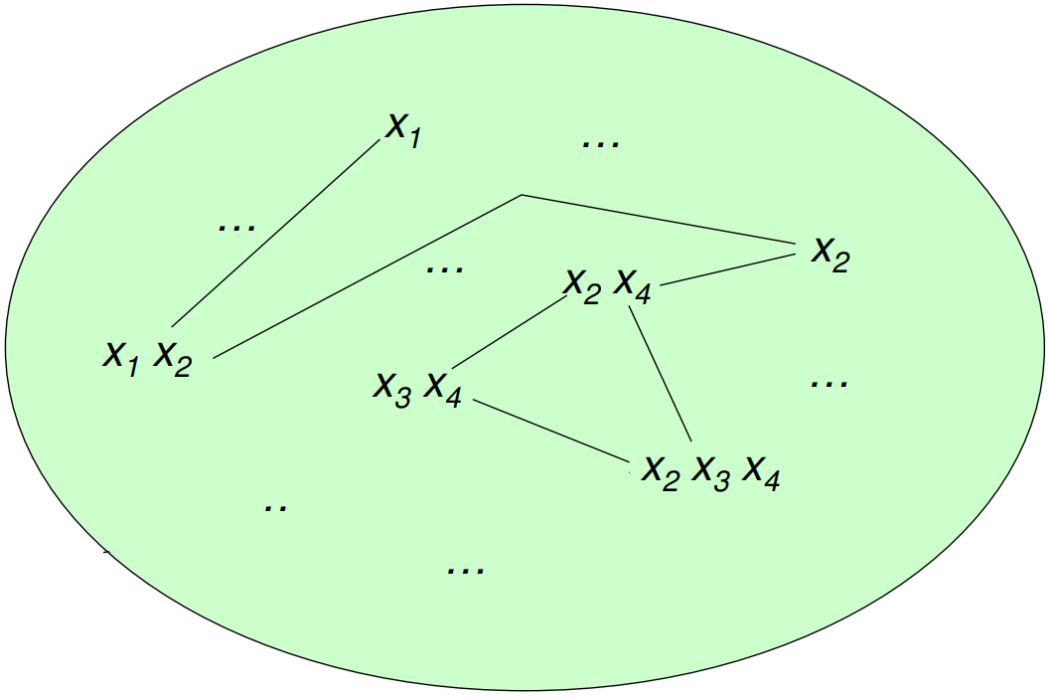
Ótimos Locais vs Ótimos Globais

- O **espaço de busca** de um problema otimização é o conjunto de todas as soluções possíveis, podendo ser restrito as soluções viáveis ou não;
- Suponha o problema da Mochila, com 4 objetos x_1, x_2, x_3, x_4 e mochila com capacidade W;



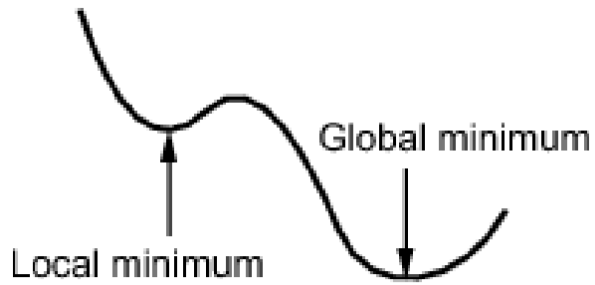
Ótimos Locais vs Ótimos Globais

- Duas soluções são vizinhas, se elas possuem elementos em comum.
- Dado uma solução viável, chamamos de **vizinhança** dessa solução, as soluções viáveis que possuem algum elementos em comum.



Ótimos Locais vs Ótimos Globais

- Um **ótimo global** ou apenas ótimo é uma solução considerada ótima entre todas as soluções viáveis possíveis.
- Ou seja, considerando-se todo o espaço de busca.
- Um **ótimo local** é uma solução considerada ótima entre todas as soluções de sua vizinhança. Ou seja, não necessariamente um ótimo global.



- De acordo com o objetivo do problema, minimização ou maximização, chamamos de mínimo local/global ou máximo local/global.

Problema de Otimização

- Temos duas Classes de problemas importantes, Polinomial e NP-Completo:
  - Informalmente, problema Polinomial são problemas que já se conhece algoritmo de tempo $O(n^k)$, $k \in R^+$: Caminho Mínimo e Árvore Geradora Mínima;
  - Informalmente, problema NP-Completo são problemas que o único algoritmo (conhecido) que resolve é o de Força Bruta, ou seja, $O(2^n)$ ou $O(n!)$: Caixeiro Viajante, Clique Máxima e Roteamento;
- Em muitas situações práticas, uma solução viável de boa qualidade já é suficiente;
- Normalmente, são situações em que não se tem muito tempo disponível: Aplicações em tempo real;
- Imagine se o problema de caminho mínimo fosse NP-Completo.
- Nesse cenário que o Projeto de Algoritmo Guloso surgiu.

Problema de Otimização

- Algoritmos para problemas de otimização tipicamente executam uma sequência de passos, com um conjunto de escolhas a cada passo;
- Dizemos que são algoritmo construtivos, ou seja, iterativos, onde a cada iteração, um elemento da solução é escolhido;
- Problema de Mochila: A cada iteração selecionamos um objeto para entrar na solução;
- Problema da Árvore Geradora Mínima: A cada iteração selecionamos uma aresta para entrar na solução;
- Para a execução quando uma solução completa é gerado.

Algoritmos Gulosos

- **Um Algoritmo Guloso sempre escolhe a melhor opção para o momento.** É que chamamos de **critério guloso**.
- Ele faz uma escolha ótima local esperando que esta o leve a uma solução ótima global.
- Critério Guloso:
  - O critério guloso é utilizado para fazer a escolha do próximo elemento da solução.
  - O critério guloso sempre estará relacionado a função objetiva (minimização ou maximização) e sempre deve vir junto com um **argumento**.

Critério Guloso - Problema da Mochila

- Escolha sempre o objeto de maior valor (Argumento: Como o objetivo é maximização, selecionar sempre o objeto mais valioso, no final terei uma solução de boa qualidade).
- Escolha o objeto de menor peso (Argumento: Dessa forma irei selecionar mais objetos, logo, a solução será boa).
- Escolha objeto de maior valor de $\frac{v_i}{w_i}$. (Argumento: Irei calcular o valor por peso de cada objeto e irei selecionar o objeto de maior valor por peso).

Protótipo de Algoritmo Guloso

Inicialmente, a solução é um conjunto vazio e a cada passo:

- Escolha um objeto para entrar na solução por meio do **critério guloso**.
- Verifique a viabilidade da solução.
- Se a escolha é viável: Atualize a solução.
- Senão: Rejeite este elemento (definitivamente).
- Se a solução é completa: Termine.

Protótipo de Algoritmo Guloso - Problema de Mochila

Inicialmente, a solução é um conjunto vazio e a cada passo:

- Escolha o melhor elemento do conjunto de opções por meio do **critério guloso**.
- Verifique se a soma dos pesos dos objetos que já estão sa solução mais do objeto escolhido é menor que a capacidade da mochila.
- Se a escolha é viável: Atualize a solução.
- Senão: Rejeite este o objeto (definitivamente).
- Se a solução é completa: Termine.

Protótipo de Algoritmo Guloso - Árvore Geradora Mínima

Inicialmente, a solução é um grafo com todos os vértices do grafo de entrada e a cada passo:

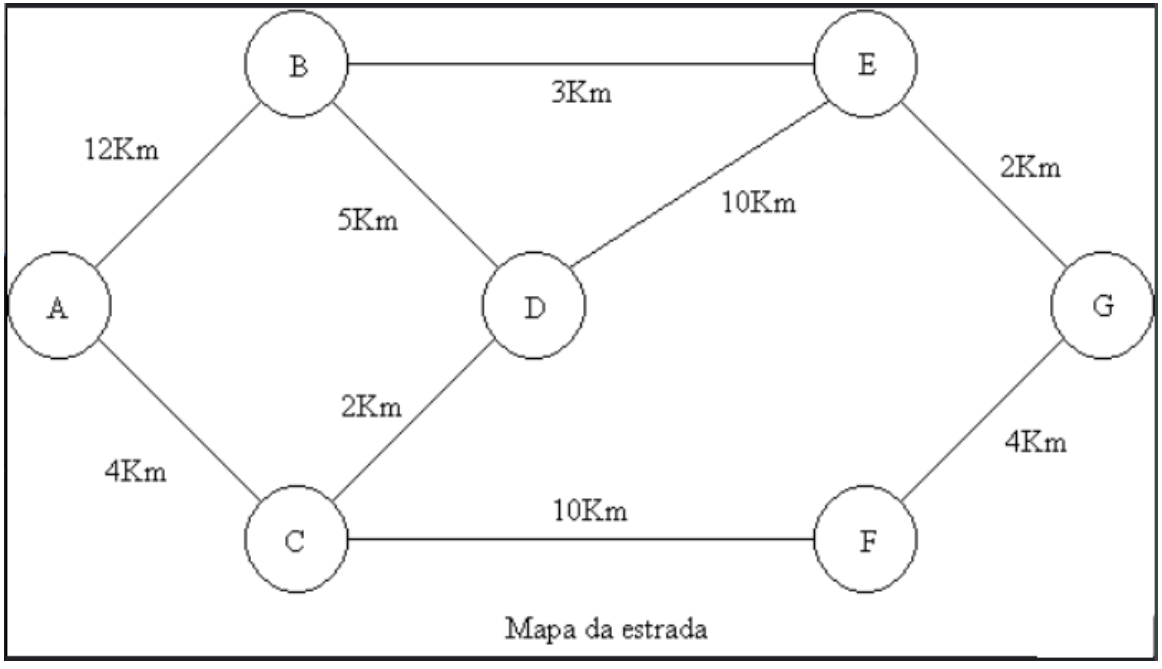
- Escolha uma aresta para entrar na solução por meio do **critério guloso**.
- Verifique se a aresta escolhida não gera um ciclo com as arestas já selecionada.
- Se a escolha é viável: Atualize a solução.
- Senão: Rejeite este o objeto (definitivamente).
- Se a solução é completa: Termine.

Algoritmos Gulosos

- Um Algoritmo Guloso é um algoritmo construtivo, ou seja, em cada iteração seleciona um elemento para entrar na solução;
- **Um Algoritmo Guloso sempre escolhe a melhor opção para o momento.** É que chamamos de **critério guloso**.
- Ele faz uma escolha ótima local esperando que esta o leve a uma solução ótima global.
- Em um algoritmo guloso uma escolha que foi feita **nunca é revista**, ou seja, não há qualquer tipo de backtracking.
- Em geral é simples fácil projetar/descrever/implementar. O difícil e provar que ele encontra o ótimo global.

Problema de Caminho Mínimo

- O problema do caminho mínimo consiste basicamente: dado um grafo com pesos nas arestas, obter o caminho de menor custo entre dois vértices.
- Qual seria um critério guloso????
- Selecione o próximo vértice do caminho que tenha a aresta de peso mínimo.



Problema da Máxima Interseção de k-subconjunto

- Dado uma coleção $L = \{S_1, S_2, \ldots, S_n\}$ de n subconjuntos de um conjunto R e um inteiro k, temos que encontrar um subconjunto $L' \subseteq L$ com $|L'| = k$ tal que a interseção desses subconjuntos sejam máxima.
- Qual seria um critério guloso????
- Selecione um subconjunto que tenha a máxima interseção com os subconjuntos já selecionados.

Problema da Máxima Interseção de k-subconjunto

Exemplo 2: $k = 3$ e
$S_1 = \{a, b, c, d, e\}, S_2 = \{a, b, d, f\}, S_3 = \{b, d, f, g\}, S_4 = \{b, d, f\}$.

- Inicialmente $L' = \emptyset$.
- Selecionamos S_1. Atualizando $L' = \{S_1\}$.
- Selecionamos S_2. Atualizando $L' = \{S_1, S_2\}$.
- Selecionamos S_3. Atualizando $L' = \{S_1, S_2, S_3\}$.
- Solução do guloso: $|S_1 \cap S_2 \cap S_3| = 2$.
- Solução ótima: $|S_4 \cap S_2 \cap S_3| = 3$.

Algoritmo Kruskal e Prim

- Ambos são algoritmos gulosos.
- O critério guloso de cada um deles.
- Kruskal: Seleciona a aresta de menor peso que não gera ciclo com as arestas já adicionadas.
- Prim: Seleciona a aresta leve(de peso mínimo) que cruza o corte.