

Algoritmos Gulosos

Conceitos e Problemas Clássicos

Prof. Me. Lucas Bruzzone

Aula 11

O que são Algoritmos Gulosos?

Definição:

- Fazem a escolha que parece melhor no momento
- Nunca reconsideram decisões anteriores
- Buscam otimização local esperando atingir o ótimo global

Característica principal:

“Escolha localmente ótima em cada etapa”

Por que estudar?

- Simples de implementar e entender
- Eficientes computacionalmente
- Funcionam para muitos problemas práticos
- Base para entender quando NÃO funcionam

Estrutura Geral de um Algoritmo Guloso

Padrão de solução:

- ① Iniciar com solução vazia
- ② **Enquanto** não tiver solução completa:
 - Selecionar próximo elemento (escolha gulosa)
 - Adicionar à solução se factível
- ③ Retornar solução construída

Elementos essenciais:

- **Função de seleção:** qual elemento escolher?
- **Função de factibilidade:** escolha é válida?
- **Função objetivo:** solução é ótima?

Quando um Algoritmo Guloso Funciona?

Propriedades necessárias:

1. Propriedade da Escolha Gulosa

- Escolha localmente ótima leva à solução globalmente ótima
- Não precisamos reconsiderar decisões anteriores

2. Subestrutura Ótima

- Solução ótima contém soluções ótimas dos subproblemas
- Similar à programação dinâmica

Diferença da PD:

- PD: Explora todas possibilidades e escolhe a melhor
- Guloso: Faz uma escolha e nunca volta atrás

Problema: Seleção de Atividades

Descrição:

- Temos n atividades com horário de início s_i e fim f_i
- Objetivo: Selecionar máximo de atividades compatíveis
- Atividades são compatíveis se não se sobrepõem

Exemplo:

Atividade	Início	Fim
A1	1	4
A2	3	5
A3	0	6
A4	5	7
A5	3	9
A6	5	9
A7	6	10
A8	8	11

Solução Gulosa: Seleção de Atividades

Estratégia gulosa: Ordenar por horário de término e sempre escolher a atividade que termina primeiro

Passo a passo:

- ① Ordenar atividades por f_i crescente
- ② Selecionar primeira atividade
- ③ Para cada atividade seguinte:
 - Se começar após término da última selecionada: incluir
 - Caso contrário: descartar

Exemplo executado:

Ordenado por fim: A1(1,4), A2(3,5), A4(5,7), A5(3,9), A6(5,9), A7(6,10),
A8(8,11)

Selecionados: A1, A4, A8 (3 atividades)

Por que a Escolha Gulosa Funciona Aqui?

Intuição:

- Escolher atividade que termina primeiro deixa mais tempo livre
- Maximiza oportunidades para escolhas futuras
- Não há vantagem em escolher atividade que termina depois

Prova por contradição (informal):

- Suponha solução ótima não usa atividade de menor fim
- Podemos substituir primeira atividade da solução ótima
- Obtemos solução tão boa quanto (ou melhor)
- Logo, escolha gulosa não piora a solução

Complexidade:

- Ordenação: $O(n \log n)$
- Seleção: $O(n)$
- **Total:** $O(n \log n)$

Problema: Mochila Fracionária

Descrição:

- Capacidade máxima da mochila: W
- n itens, cada um com peso w_i e valor v_i
- Pode levar frações de itens
- Objetivo: Maximizar valor total

Exemplo:

Item	Peso	Valor	Valor/Peso
1	10 kg	60	6,0
2	20 kg	100	5,0
3	30 kg	120	4,0

Capacidade: 50 kg

Solução Gulosa: Mochila Fracionária

Estratégia gulosa: Ordenar por valor por unidade de peso (v_i/w_i) e pegar itens na ordem decrescente

Algoritmo:

- ① Calcular v_i/w_i para cada item
- ② Ordenar por essa razão (decrescente)
- ③ Pegar itens na ordem:
 - Se couber inteiro: pegar tudo
 - Se não couber: pegar fração que cabe

Exemplo executado ($W = 50$):

- Item 1: 10 kg completo \rightarrow valor = 60
- Item 2: 20 kg completo \rightarrow valor = 100
- Item 3: 20 kg (2/3 do item) \rightarrow valor = 80
- **Total: 240** (máximo possível)

Por que Funciona na Versão Fracionária?

Intuição:

- Queremos maximizar valor por kg usado
- Pegar item de maior razão é sempre ótimo
- Frações permitem aproveitar totalmente a capacidade

Propriedade chave:

- Não há penalidade por dividir itens
- Cada fração contribui proporcionalmente
- Escolha local (maior razão) = escolha global ótima

Complexidade:

- Calcular razões: $O(n)$
- Ordenação: $O(n \log n)$
- Seleção: $O(n)$
- **Total:** $O(n \log n)$

Vantagens dos Algoritmos Gulosos

Quando funcionam, são excelentes:

- **Simples:** Fáceis de implementar e entender
- **Eficientes:** Geralmente $O(n \log n)$ ou melhor
- **Sem backtracking:** Menor uso de memória
- **Intuitivos:** Estratégia natural para muitos problemas

Comparação com outras técnicas:

Técnica	Tempo típico	Espaço
Força bruta	$O(2^n)$ ou $O(n!)$	$O(n)$
Programação Dinâmica	$O(n^2)$ a $O(n^3)$	$O(n^2)$
Guloso	$O(n \log n)$	$O(1)$ a $O(n)$

Limitações dos Algoritmos Gulosos

Principal problema: Nem sempre produzem solução ótima

Quando falham:

- Escolha local ótima pode bloquear escolhas melhores futuras
- Falta de visão global do problema
- Ausência das propriedades necessárias

Exemplo clássico de falha:

- Mochila 0/1 (não pode dividir itens)
- Problema do troco com certas moedas
- Caminho mais longo em grafo

Conclusão:

É necessário provar que a estratégia gulosa funciona!

Como Saber se Guloso Funciona?

Abordagem prática:

① Verificar propriedades:

- Escolha gulosa leva ao ótimo?
- Existe subestrutura ótima?

② Tentar contra-exemplos:

- Criar instâncias onde guloso falha
- Se falhar, guloso não funciona

③ Prova formal:

- Demonstrar que estratégia é ótima
- Geralmente por indução ou contradição

Importante:

- Na dúvida, implemente e teste exaustivamente
- Compare com força bruta em casos pequenos
- Se não conseguir provar, provavelmente não funciona

Exercícios para Praticar

1. Troco com moedas canônicas

- Moedas: {1, 5, 10, 25, 50, 100}
- Dar troco de 63 centavos usando mínimo de moedas

2. Minimizar tempo de espera

- Tarefas com durações: [3, 7, 2, 5]
- Processar todas em uma máquina
- Qual ordem minimiza tempo médio de espera?

3. Carregar contêineres

- Capacidade: 15 kg
- Pesos: [5, 3, 7, 2, 8, 4]
- Maximizar quantidade de contêineres

Tema: Quando Algoritmos Gulosos Falham

O que veremos:

- Problema do troco com moedas não-canônicas
- Mochila 0/1: por que guloso não funciona
- Exemplos de contra-exemplos
- Como reconhecer quando guloso vai falhar
- Introdução à necessidade de outras técnicas (PD)

Para casa:

- Implementar seleção de atividades
- Resolver os 3 exercícios propostos
- Pensar: por que cada estratégia gulosa funciona?