AULA 14 – ALGORITMOS GULOSOS

Prof. Daniel Kikuti

Universidade Estadual de Maringá

1 de outubro de 2014

Conteúdo

- Elementos de algoritmos gulosos.
- Um problema de escalonamento de tarefas.
- Definindo a escolha gulosa.
- Subestrutura ótima.
- Exercícios.

Elementos de algoritmos gulosos

Subestrutura ótima

Um subproblema exibe **subestrutura ótima** se uma solução ótima para um problema contém dentro dela soluções ótimas para subproblemas.

Escolha gulosa

Podemos construir uma solução ótima global fazendo escolhas ótimas locais (gulosas – sem considerar os resultados dos subproblemas).

O problema de escalonamento de tarefas

O problema

Dado um conjunto de n **tarefas** $S = \{1, 2, \cdots, n\}$ que requerem o uso exclusivo de um recurso comum (processador, por exemplo).
Cada tarefa i possui:

- **comprimento** t_i (tempo necessário para executá-la);
- **prioridade** p_i (ou peso).

Definimos o tempo de término c_i para a tarefa i como sendo a soma de todos os tempos de término das tarefas antecedentes a i, incluindo t_i.
Objetivo: Minimizar a soma ponderada dos tempos de término:

$$\min \sum_{i=1}^{n} p_i c_i.$$

Exemplo

Suponha que temos 3 tarefas com os seguintes comprimentos e pesos:

- $t_1 = 1, \; t_2 = 2, \; t_3 = 3.$
- $p_1 = 3, \; p_2 = 2, \; p_3 = 1.$

Pergunta 1

Se as tarefas são escalonadas na ordem $\{1, 2, 3\}$, quais são os tempos de término de c_1, c_2 e c_3?

Pergunta 2

Qual o valor da soma ponderada?

Pergunta 3

De quantas maneiras distintas podemos escalonar n tarefas?

Exemplo

Suponha que temos 3 tarefas com os seguintes comprimentos e pesos:

- $t_1 = 1$, $t_2 = 2$, $t_3 = 3$.
- $p_1 = 3$, $p_2 = 2$, $p_3 = 1$.

Pergunta 1

Se as tarefas são escalonadas na ordem $\{1, 2, 3\}$, quais são os tempos de término de c_1, c_2 e c_3?
Resposta: 1, 3 e 6.

Pergunta 2

Qual o valor da soma ponderada?
Resposta: $3 * 1 + 2 * 3 + 1 * 6 = 15$.

Pergunta 3

De quantas maneiras distintas podemos escalonar n tarefas?
Resposta: $n!$ maneiras.

Analisando a relação tempo vs. prioridade

$$\min \sum_{i=1}^{n} p_i c_i.$$

1. Se todas as tarefas possuem a mesma prioridade, qual a melhor maneira de escalonar as tarefas?
   Exemplo: $t_1 = 1$, $t_2 = 2$, $t_3 = 3$ e $p_i = 1$.

2. E se todas as tarefas possuem o mesmo comprimento, qual a melhor maneira de escalonar as tarefas?
   Exemplo: $p_1 = 3$, $p_2 = 2$, $p_3 = 1$ e $t_i = 2$.

Analisando a relação tempo vs. prioridade

$$\min \sum_{i=1}^{n} p_i c_i.$$

1. Se todas as tarefas possuem a mesma prioridade, qual a melhor maneira de escalonar as tarefas?
   Exemplo: $t_1 = 1$, $t_2 = 2$, $t_3 = 3$ e $p_i = 1$.
   Menor tempo primeiro.
2. E se todas as tarefas possuem o mesmo comprimento, qual a melhor maneira de escalonar as tarefas?
   Exemplo: $p_1 = 3$, $p_2 = 2$, $p_3 = 1$ e $t_i = 2$.
   Maior prioridade primeiro.

Resolvendo conflito

Caso geral

E se $p_i > p_j$ e $t_i > t_j$? Qual tarefa deve ser escalonada primeiro?

Ideia

Atribuir uma pontuação para cada tarefa de modo que a tarefa com maior prioridade e menor comprimento receba uma pontuação maior.

Sugestões de funções de pontuação

Resolvendo conflito

Caso geral

E se $p_i > p_j$ e $t_i > t_j$? Qual tarefa deve ser escalonada primeiro?

Ideia

Atribuir uma pontuação para cada tarefa de modo que a tarefa com maior prioridade e menor comprimento receba uma pontuação maior.

Sugestões de funções de pontuação

1. Colocar as tarefas em ordem decrescente conforme a pontuação $p_i - t_i$.
2. Colocar as tarefas em ordem decrescente conforme a pontuação p_i / t_i.
3. ???

Analisando funções de pontuação

Suponha $t_1 = 5$, $p_1 = 3$ e $t_2 = 2$, $p_2 = 1$.
A soma ponderada dos tempos de conclusão produzidos pelas funções de pontuação são:

Analisando funções de pontuação

Suponha $t_1 = 5$, $p_1 = 3$ e $t_2 = 2$, $p_2 = 1$.
A soma ponderada dos tempos de conclusão produzidos pelas funções de pontuação são:

1. Função $p_i - t_i$

   - pontuação da tarefa 1: $3 - 5 = -2$.
   - pontuação da tarefa 2: $1 - 2 = -1$.

Portanto, executar tarefa 2 antes da tarefa 1. Custo total será $2 * 1 + 7 * 3 = 23$.

Analisando funções de pontuação

Suponha $t_1 = 5$, $p_1 = 3$ e $t_2 = 2$, $p_2 = 1$.
A soma ponderada dos tempos de conclusão produzidos pelas funções de pontuação são:

1. Função $p_i - t_i$
   - pontuação da tarefa 1: $3 - 5 = -2$.
   - pontuação da tarefa 2: $1 - 2 = -1$.

Portanto, executar tarefa 2 antes da tarefa 1. Custo total será $2 * 1 + 7 * 3 = 23$.

2. Função p_i / t_i
   - pontuação da tarefa 1: $3/5 = 0.6$.
   - pontuação da tarefa 2: $1/2 = 0.5$.

Portanto, executar tarefa 1 antes da tarefa 2. Custo total será $5 * 3 + 7 * 1 = 22$.

Escolha gulosa

Teorema

A escolha de tarefas ordenadas em ordem decrescente conforme a razão p_i/t_i está sempre correta.

Demonstração

- Seja δ o escalonamento guloso e δ^* um escalonamento ótimo.
- Assumiremos que todos p_i/t_i são distintos e renomearemos as tarefas de forma que:

$$p_1/t_1 > p_2/t_2 > \cdots > p_{n-1}/t_{n-1} > p_n/t_n.$$

- Então o escalonamento guloso será: $\delta = \{1, 2, \ldots, n\}$.

Escolha gulosa

Demonstração – continuação

Suponha que $\delta \neq \delta^*$, então existem tarefas consecutivas i e j com $i > j$ com posições invertidas. Se trocarmos a ordem de i e j em δ^* (mantendo as outras tarefas inalteradas), percebemos que:

- o tempo de conclusão de qualquer outra tarefa k permanece inalterado;
- o tempo de conclusão da tarefa i aumenta t_j unidades;
- o tempo de conclusão da tarefa j diminui t_i unidades;

Então o tempo de conclusão ponderado seria:

$$\sum_{k=1}^{i-1} p_k c_k + p_i(c_i + t_j) + p_j(c_j - t_i) + \sum_{k=j+1}^{n} p_k c_k.$$

Mas $i > j \Rightarrow \frac{p_i}{t_i} < \frac{p_j}{t_j} \Rightarrow p_i t_j < p_j t_i$, ou seja, o benefício é maior que o custo e portanto é possível melhorar δ^*, o que contradiz a otimalidade de δ^*.

Concluindo

Subestrutura ótima

Se a tarefa com maior p_i/t_i for removida da solução ótima, então a solução restante para $n - 1$ tarefas é ótima.

Exercícios

- Mostre que o problema possui subestrutura ótima.
- Escrever o algoritmo para este problema e analisar sua complexidade.