Algoritmos

Pedro Hokama

Fontes

- [*clrs*] Algoritmos: Teoria e Prática (Terceira Edição) Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest e Clifford Stein.
- [*timr*] Algorithms Illuminated Series, Tim Roughgarden
- Desmistificando Algoritmos, Thomas H. Cormen.
- Algoritmos, Sanjoy Dasgupta, Christos Papadimitriou e Umesh Vazirani
- Stanford Algorithms
  `https://www.youtube.com/playlist?list=PLXFMml k03Dt7Q0xr1PIAriY5623cKiH7V`
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt5EMI2s2WQBsLsZl7A5HEK6`
- Conjunto de Slides dos Professores Cid C. de Souza, Cândida N. da Silva, Orlando Lee, Pedro J. de Rezende
- Conjunto de Slides do Professores Cid C. de Souza para a disciplina MO420

Qualquer erro é de minha responsabilidade.

.

- Uma construtora lucra 4.6 milhões com a venda de um prédio comercial e 2.2 milhão com a venda de casas de luxo.
- Para construir um prédio comercial a construtora aloca 40 funcionários, e para as casas de luxo, 19 funcionários. Sendo que a construtora tem um total de 100 funcionários. As duas obras gastam o mesmo tempo.
- Em quantos prédio comerciais e casas de luxo a construtora deve investir?

Em quantos prédio comerciais e casas de luxo a construtora deve investir?
x é o número de prédios.
y é o número de casas.

$$\begin{aligned} \text{maximizar} \quad & 4.6x + 2.2y, \\ \text{sujeito a} \quad & 40x + 19y \leq 100, \\ & x \geq 0, \quad y \geq 0. \end{aligned}$$

$$\begin{aligned} \text{maximizar} \quad & 4.6x + 2.2y, \\ \text{sujeito a} \quad & 40x + 19y \leq 100, \\ & x \geq 0, \quad y \geq 0. \end{aligned}$$

- A solução ótima para esse modelo é $x = 0$ e $y = 5.26$ que dá um lucro de 11.58 milhões.
- Infelizmente construir 26% de uma casa não dá lucro algum, de fato pode até trazer algum prejuízo.

- Podemos arredondar para baixo (5 casas) porém essa solução não é mais ótima.
- Precisamos que as variáveis sejam inteiras!

$$\begin{aligned} \text{maximizar} \quad & 4.6x + 2.2y, \\ \text{sujeito a} \quad & 40x + 19y \leq 100, \\ & x \geq 0, \quad y \geq 0. \\ & x, y \in \mathbb{Z} \end{aligned}$$

- A solução ótima desse modelo é $x = 2$ e $y = 1$.

Programação Linear Inteira (PLI)

$$\begin{aligned} \text{maximizar} \quad & 4.6x + 2.2y, \\ \text{sujeito a} \quad & 40x + 19y \leq 100, \\ & x \geq 0, \quad y \geq 0. \\ & x, y \in \mathbb{Z} \end{aligned}$$

- Quando tempos variáveis inteiras, utilizamos a Programação Linear Inteira - PLI (Integer Linear Programming - ILP).

- Quando temos algumas variáveis inteiras e outras continuas temos a Programação Linear Mista (Mixed-Integer Programming - MIP).
- Infelizmente resolver um PLI ou MIP é NP-Difícil.
- Para mostrar que resolver um PLI é um problema NP-Difícil vamos mostrar uma redução do problema da mochila para para um PLI.

Problema da Mochila

Dado uma coleção I de n itens e uma capacidade inteira W. Cada item $i \in I$ tem:

- Um valor v_i (não negativo)
- Um peso w_i (não negativo e inteiro)

Encontrar $S \subseteq I$ cujo peso não ultrapasse W, ou seja,

$$\sum_{i \in S} w_i \leq W$$

e que maximiza $\sum_{i \in S} v_i$.

Formulações em PLI

Atenção

As formulações dos problemas apresentados a seguir foram feitas na lousa em sala de aula.

Formulações em PLI

- A vantagem de usar PLI é que temos todo o framework já desenvolvido para resolve-los.
- De fato como PLI é NP-Difícil poderíamos transformar qualquer problema em NP em um Programa Linear Inteiro.
- Não existe uma formula simples que funcione para qualquer problema, mas com um pouco de experiência e criatividade é possível formular uma grande variedade de problemas.

Problema de Alocação de Pessoal

São dados n pessoas e n tarefas. Cada pessoa exercerá uma tarefa, e é conhecido o valor c_{ij} indica a adequação da pessoa i para realizar a tarefa j. O objetivo é encontrar uma alocação que maximize a soma da adequação.

Problema de Coloração em Grafos

Dado um grafo não direcionado $G = (V, E)$, queremos encontrar uma cor para cada vértice, de forma que dois vértices adjacentes não seja tenham a mesma cor. O objetivo é minimizar o número de cores utilizadas.

Problema de Virtualização de Máquinas

São dados um conjunto de tarefas S, cada tarefa $s \in S$ com um tempo de processamento t_s, e um tempo limite C. Cada tarefa precisa ser processada em apenas uma máquina, mas cada máquina processa várias tarefas (somando seus tempos), todas as tarefas precisam ser processadas até C.
Desejamos minimizar a quantidade necessária de máquinas virtuais.

- O problema anterior é conhecido como Bin-Packing, ou empacotamento de contêiner, ou ainda como um problema de Corte.
- Aparece com frequência em diferentes contextos, corte de bobinas de papel, vergalhões, madeira, empacotamento de contêineres para exportação, além de situações mais abstratas como a de Virtualização de Máquinas, Alocação de Recursos Financeiros, etc etc.

Problema de Localização de Instalações

São dados um conjunto F de possíveis instalações e um conjunto C de clientes, sendo:

- f_j o custo de abrir a instalação $j \in F$
- d_{ij} o custo de ligar o cliente $i \in C$ na instalação $j \in F$

O objetivo é encontrar quais instalações abrir e conectar todo cliente a uma instalação aberta minimizando o custo total.