Segunda Prova - Otimização
Prof. André Guedes
8 de agosto de 2024

1. (10 pontos) Temos um problema P que pode ser modelado como um problema de programação linear inteira (PLI) de maximização. Queremos resolver P usando a técnica de *Branch & Bound* (B&B). Como podemos usar a modelagem de P como PLI para encontrar uma função limitante a ser usada no algoritmo B&B? O que podemos dizer sobre a "qualidade" desta função limitante?

2. (10 pontos) O que é e como funciona um *Algoritmo Guloso*? Explique com exemplos.

3. (10 pontos) Considere um certo problema de otimização P e um algoritmo de *Branch & Bound* (B&B) A e seu esquema de ramificação. Suponha que as soluções viáveis de P são sequências (ordenadas) sobre um conjunto dado S. Assuma que temos duas opções para a ramificação em A: binária, usando o vetor característico dos subconjuntos de S; ou não binária, usando um elemento de S para cada filho. Qual delas você usaria? Ou usaria uma terceira opção? Justifique?

4. Considere a seguinte recorrência que descreve as soluções de um problema Q em função de subproblemas:

$$Q(S,k) = \begin{cases} 0, & \text{se } S = \emptyset \text{ ou } k = 0, \\ \max\{Q(S \setminus x, k - v(x)) + |S|v(x) \mid x \in S, v(x) \leq k\} \cup \{0\}, & \text{caso contrário.} \end{cases}$$

   Onde, queremos resolver o problema $Q(U, C)$, U é um conjunto com $n > 0$ elementos, C é um número inteiro não negativo, e v é uma função de U para os naturais. Podem assumir que $\sum_{x \in U} v(x) > C$. Responda:

   (a) (10 pontos) Os subproblemas se repetem?

   (b) (15 pontos) É possível usar a técnica de *Backtracking* para resolver Q? É preciso alguma adaptação? Qual seria? Como seria a ramificação e a escolha de candidatos?

   (c) (15 pontos) É possível usar a técnica de *Branch & Bound* para resolver Q? É preciso alguma adaptação? Qual seria? Como poderia ser uma função limitante?

   (d) (15 pontos) É possível usar a técnica de *Programação Dinâmica* para resolver Q? É preciso alguma adaptação? Qual seria? Qual seria a estrutura de dados usada para guardar os subproblemas resolvidos e qual seria a ordem de execução?

   (e) (15 pontos) É possível usar a técnica de *Algoritmos Gulosos* para resolver Q? É preciso alguma adaptação? Qual seria?[1]

[1] Considere que a existência de tal algoritmo está sempre condicionada a existência ou não de uma função "oráculo". Você deve responder assumindo que tal função existe.

GABARITO

1. Podemos usar a versão relaxada do PLI como função limitante, pois sempre teremos que seu valor será sempre maior ou igual ao PLI original. Entretanto, de forma geral, não podemos falar nada sobre a qualidade desta função limitante, já que em certos problemas ela pode ser bem próxima ao valor ótimo e em outros pode ser arbitrariamente distante.

2. Um algoritmo guloso pode ser pensado como uma estratégia de construção "passo a passo" da solução sem que seja necessário reavaliar. Ou seja, cada variável da solução ótima é decidida, uma a uma. Uma outra forma é pensar que em uma árvore de recursão (com escolhas), temos uma escolha de qual ramo seguir, sem voltar atrás. Um exemplo seria ...

3. Como a resposta é uma sequência, a ramificação binária, baseada no vetor característico, não serve, pois não consegue representar a ordem da sequência, só subconjuntos (sem ordem). Portanto a ramificação adequada é a não binária, com um filho para cada elemento de S.

4. (a) Sim! Escolhendo a depois b gera o mesmo subproblema que escolher b depois a, por exemplo.
   (b) Sim, sem modificações. Basta ramificar com um filho para cada elemento $x \in S$ tal que $v(x) \leq k$.
   (c) Sim, sem modificações. Uma função limitante poderia ser o acumulado (até o momento) somado com $M(|S|(|S|+1)/2)$, onde M é o máximo valor de $v(x)$ tal que $x \in S$ e $v(x) \leq k$. Esta função estia o valor ótimo considerando que todos os elementos x de S são tais que $v(x) \leq k$ e são iguais ao máximo. (ESSE É SÓ UM EXEMPLO!)
   (d) Sim, sem modificações. A estrutura de dados deve ser uma matriz indexada por subconjuntos de U e $0 \leq k \leq C$, ou seja com dimensões $2^{|U|} \times C + 1$. A ordem seria fazer primeiro as linhas com subconjuntos menores.
   (e) Caso exista uma função oráculo, podemos usar a estratégia gulosa, pois a recorrência que resolve o problema tem escolhas.