Prova Final - Otimização
Prof. André Guedes
15 de agosto de 2024

1. (10 pontos) É correto afirmar que o Simplex utiliza uma estratégia gulosa? Quais diferenças você pode citar em relação a algum dos algoritmo gulosos vistos em sala de aula (por exemplo, para resolver o problema da mochila fracionária)?

2. Para cada item, apresente um exemplo de programa linear com as características descritas.
   (a) (5 pontos) É inviável.
   (b) (5 pontos) É ilimitado.
   (c) (5 pontos) Tem uma única solução.
   (d) (5 pontos) Tem infinitas soluções.

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

$$R(S,i) = \begin{cases} 0, & \text{se } i = 0 \text{ ou } S = \emptyset, \\ \max\{a_i x + R(S \setminus x, i-1) \mid x \in S\}, & \text{caso contrário.} \end{cases}$$

   Onde queremos resolver o problema $R(U,n)$, $U \subset \mathbb{N}$ é um conjunto com $m \geq n$ elementos e $a_i \in \mathbb{R}$, com $a_i \geq 0$, $i = 1..n$, são constantes dadas.

   (a) (15 pontos) Apresente uma descrição do problema R, apresentando quais são as variáveis, as restrições e a função objetivo. O problema R pode ser classificado como um problema de Programação Linear ou Programação Linear Inteira? Justifique.
   (b) (15 pontos) Explique como a técnica de *Backtracking* pode ser usada para resolver R. Como deve ser feita a ramificação e escolha de candidatos?

4. (15 pontos) Em um algoritmo de *Branch and Bound*, qual a diferença entre cortes por viabilidade e cortes por otimalidade? Apresente um exemplo. Dado um problema de otimização P, quais estratégias podem ser empregadas para efetuar tais cortes?

5. Considere um jogo no qual n cartas são colocadas na mesa com a face voltada para cima e cada um de dois jogadores deve remover, em sua rodada, ou a primeira ou a última carta, com o objetivo de maximizar a soma das cartas removidas. Supondo que ambos os jogadores jogam de maneira ótima, considere o problema de encontrar o valor máximo da soma $J_1 - J_2$, onde J_i é a soma das cartas removidas pelo i-ésimo jogador; isto é, a maior vantagem que o primeiro jogador pode obter sobre o segundo. Temos a seguinte sub-estrutura ótima:

$$P(i,j) = \begin{cases} 0, & \text{se } i > j, \\ \max\{c_i - P(i+1,j), c_j - P(i,j-1)\}, & \text{caso contrário.} \end{cases}$$

   Onde queremos resolver o problema $P(1,n)$, n par, sendo $c_1, c_2, \cdots, c_n$ os valores das cartas sobre a mesa. Note que a subtração se refere a alternância de jogadores.

   (a) (15 pontos) Considere a estratégia gulosa de sempre pegar a maior carta (dentre as duas opções). Você acha que isso funciona? Tem outra proposta de estratégia gulosa?
   (b) (15 pontos) Nesta recorrência há repetição de subproblemas? É possível usar Programação Dinâmica (PD) para resolver este problema usando esta recorrência? Como deve ser a estrutura de dados usadas nesta PD?

GABARITO

1. Não! O Simplex não está "construindo" a solução a cada passo. Ao invés disso está "percorrendo" o espaço de busca.

2. (a)

$$\begin{aligned}&\max x\\&\text{s.t.}\\&x \leq 2\\&x \geq 4\end{aligned}$$

(b)

$$\begin{aligned}&\max x\\&\text{s.t.}\\&x \geq 4\end{aligned}$$

(c)

$$\begin{aligned}&\max x\\&\text{s.t.}\\&x \leq 2\\&x \geq 0\end{aligned}$$

(d)

$$\begin{aligned}&\max x\\&\text{s.t.}\\&x \leq 2\\&y \leq 2\\&x \geq 0\\&y \geq 0\end{aligned}$$

3. (a) Uma solução é uma sequência $(x_1, x_2, \ldots, x_n)$ tal que $x_i \in U$ e $x_i \neq x_j$ se $i \neq j$, e que maximiza $a_1x_1 + a_2x_2 + \cdots + a_nx_n$. Então, as variáveis são $x_1, \ldots, x_n$; a função objetivo é $\max a_1x_1 + a_2x_2 + \cdots + a_nx_n$ e as restrições são: $x_i \in U$ para todo i, $x_i \neq x_j$ para todo par $i \neq j$. A formulação como PLI pode ser vista abaixo.

$$\begin{aligned}&\max a_1x_1 + a_2x_2 + \cdots + a_nx_n\\&\text{s.t.}\\&x_i \in U \quad \forall i \in \{1..n\}\\&x_i \neq x_j \quad \forall i \neq j \in \{1..n\}\end{aligned}$$

(b) Ramificação para cada elemento de U ainda não escolhido.

4. ...

5. (a) Não. Se temos as cartas $(1, 7, 10, 2)$, a estratégia proposta vai fazer o jogador 1 pegar as cartas 2 e 7, somando 9, enquanto jogador 2 pega as cartas 10 e 1, somando 11. Assim a diferença seria $9 - 11 = -2$. Entretanto a diferença máxima seria atingida quando o jogador 1 pega as cartas 1 e 10 e o jogador 2 pega 7 e 2, dando $11 - 9 = 2$. ...

(b) Os problemas se repetem. Sim, podemos usar PD. A estrutura pode ser uma matriz $n \times n$ onde a posição $[i, j]$ representa o resultado de $P(i, j)$. Toda a parte inferior da diagonal principal é inicializada com 0 (zero), pois é o caso base. A ordem de execução deve ser da diagonal principal em direção para direita e acima. O custo de calcular cada posição é $O(1)$, e portanto o custo total é $O(n^2)$.