Soluções prova 2

Questão 0.1 (Formulação)
Vamos primeira formular a solução *sem* minimizar o máximo, que é simples. Seja $x_i \in \mathbb{B}$ uma variável que indica os distritos selecionados.

$$\begin{array}{lll} \textbf{min} & \sum_{1\leq i\leq N} x_i c_i & \\ \textbf{s.a} & \sum_{1\leq j\leq N} a_{ij}x_j = 1, \forall 1 \leq i \leq M & \text{Seleciona cada município uma vez} \end{array}$$

O problema da formulação é minimizar o máximo dos c_i selecionados. Para esse fim vamos introduzir variáveis $m_i \in \mathbb{B}$, $1 \leq i \leq N$ que indicam o distrito selecionado acom *população máxima*:

$$\begin{array}{lll} \textbf{min} & \sum_{1\leq i\leq N} m_i c_i & \\ \textbf{s.a} & \sum_{1\leq j\leq N} a_{ij}x_j = 1, \forall 1 \leq i \leq N & \text{Seleciona cada município uma vez} \\ & \sum_{1\leq i\leq N} m_i = 1 & \text{Seleciona exatamente um máximo} \\ & m_i \leq x_i, \forall 1 \leq i \leq N & \text{Máximo tem que ser selecionado} \\ & c_i x_i \leq c_j m_j + K(1-m_j), \forall 1 \leq i,j \leq N & \text{Máximo é máximo mesmo} \end{array}$$

A constante K pode ser definido por $K = \max_{1\leq i\leq N} c_i$.

Questão 0.2 (Formulação)
Seja $x_{ijk} \in \mathbb{B}$, $1 \leq i,j,k \leq n$ um indicador que posição (i,j) do tabuleiro contém símbolo k.

$$\begin{array}{lll} \textbf{max} & \sum_{1\leq i,k\leq n} x_{iik}k & \\ \textbf{s.a} & \sum_{1\leq k\leq n} x_{ijk} = 1 & \text{Só um símbolo por posição} \\ & \sum_{1\leq j\leq n} x_{ijk} = 1, 1 \leq i,k \leq n & \text{Linha } i \text{ contém símbolo } k \text{ 1 vez} \\ & \sum_{1\leq i\leq n} x_{ijk} = 1, 1 \leq j,k \leq n & \text{Coluna } j \text{ contém símbolo } k \text{ 1 vez} \\ & x_{1jk} = \begin{cases} 1 & \text{caso } j = k \\ 0 & \text{caso contrário} \end{cases} & \text{Forma padrão linha 1} \\ & x_{i1k} = \begin{cases} 1 & \text{caso } i = k \\ 0 & \text{caso contrário} \end{cases} & \text{Forma padrão coluna 1} \end{array}$$

Questão 0.3 (Resolução de programas inteiras)
A matriz de coeficientes é

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

é possui determinante 2, é portanto não é totalmente unimodular. A solução da relaxação linear gera a seguinte seqüência de dicionários.
Dicionário inicial:

$$\begin{array}{rrrrr} z = & 0 & +x_1 & +x_2 & +x_3 \\ x_4 = & 1 & -x_1 & -x_2 & \\ x_5 = & 1 & & -x_2 & -x_3 \\ x_6 = & 1 & -x_1 & & -x_3 \end{array}$$

Após pivô x_1–x_4

$$\begin{array}{rrrrr} z = & 1 & -x_4 & & +x_3 \\ x_1 = & 1 & -x_4 & -x_2 & \\ x_5 = & 1 & & -x_2 & -x_3 \\ x_6 = & 0 & +x_4 & +x_2 & -x_3 \end{array}$$

Após pivô x_3–x_6

$$\begin{array}{rrrrr} z = & 1 & & +x_2 & -x_6 \\ x_1 = & 1 & -x_4 & -x_2 & \\ x_5 = & 1 & -x_4 & -2x_2 & +x_6 \\ x_3 = & 0 & +x_4 & +x_2 & -x_6 \end{array}$$

Após pivô x_2–x_5 temos a solução

$$\begin{array}{rrrrr} z = & 3/2 & -1/2x_4 & -1/2x_5 & -1/2x_6 \\ x_1 = & 1/2 & -1/2x_4 & +1/2x_5 & -1/2x_6 \\ x_2 = & 1/2 & -1/2x_4 & -1/2x_5 & +1/2x_6 \\ x_3 = & 1/2 & +1/2x_4 & -1/2x_5 & -1/2x_6 \end{array} \tag{1}$$

que não é inteira. Logo, temos que aplicar outra técnica para obter a solução inteira.

Resolução com cortes de Gomory Colocamos o corte associado com a variável fracionário x_1 no sistema para obter

$$\begin{array}{rrrrr} z = & 3/2 & -1/2x_4 & -1/2x_5 & -1/2x_6 \\ x_1 = & 1/2 & -1/2x_4 & +1/2x_5 & -1/2x_6 \\ x_2 = & 1/2 & -1/2x_4 & -1/2x_5 & +1/2x_6 \\ x_3 = & 1/2 & +1/2x_4 & -1/2x_5 & -1/2x_6 \\ x_7 = & -1/2 & +1/2x_4 & +1/2x_5 & +1/2x_6 \end{array}$$

e após o pivô x_4–x_7 obtemos a solução ótima

$$\begin{array}{rrrrr} z = & 1 & -x_7 & & \\ x_1 = & 0 & -x_7 & +x_5 & \\ x_2 = & 0 & -x_7 & & +x_6 \\ x_3 = & 1 & +x_7 & -x_5 & -x_6 \\ x_4 = & 1 & +2x_7 & -x_5 & -x_6 \end{array}$$

Resolução com Branch-and-bound O dicionário final de relaxação linear forma a raiz da árvore do Branch/and/bound. Como todas variáveis possuem valor 1/2 o critério que escolhe a variável mais fracionária para ramificar não define uma preferência, portanto vamos ramificar pela variável x_1. No primeiro ramo, colocamos a nova restrição

$$\begin{array}{ll} x_1 \leq 0 & \\ x_7 = 0 - x_1 & \text{com variável de folga} \\ x_7 = -1/2 + 1/2x_4 - 1/2x_5 + 1/2x_6 & \text{re-escrito com variáveis nulas} \end{array}$$

no dicionário final 1 e a re-otimização com um pivô x_4–x_7 obtemos uma solução inteira

$$\begin{array}{rllll} z = & 1 & & & \\ x_1 = & 0 & -x_7 & & \\ x_2 = & 0 & -x_7 & & -x_5 \\ x_3 = & 1 & +x_7 & & \\ x_4 = & 1 & +2x_7 & +x_5 & -x_6 \end{array}$$

No segundo ramo, colocamos a nova restrição

$$\begin{array}{rll} x_1 \geq 1 & & \\ -x_1 \leq -1 & & \text{em forma padrão} \\ x_7 = -1 - x_1 & & \text{com variável de folga} \\ x_7 = -3/2 + 1/2x_4 - 1/2x_5 + 1/2x_6 & & \text{re-escrito com variáveis nulas} \end{array}$$

e a re-otimização com o pivô x_5–x_7 resulta em outra solução inteira

$$\begin{array}{rllll} z = & 1 & -x_4 & -x_7 & -x_6 \\ x_1 = & 1 & -x_4 & +x_7 & \\ x_2 = & 0 & -x_4 & -x_7 & \\ x_3 = & 1 & & -x_7 & -x_6 \\ x_5 = & 1 & +x_4 & +2x_7 & +x_6 \end{array}$$

Em resumo, o algoritmo Branch-and-bound explora dois ramos, que levam imediatamente em duas soluções inteiras diferentes, sem aplicar nenhum corte.

Questão 0.4 (Resolução de programas inteiras)
A matriz de coeficientes é

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}.$$

O critério de partição no pode ser aplicado diretamente, porque a matriz possui colunas com mais que dois coeficientes não-nulas. Mas, o critério se aplica para matriz transposta, i.e. às colunas. Particionando as colunas em uma partição com as colunas 1 e 2, e outra com as colunas 3 e 4, o critério de partição é satisfeito. Portanto a matriz de coeficientes é TU e a relaxação inteira possui solução inteira. Portanto, para resolver a questão, e suficiente calcular a solução da relaxação inteira, como segue:
O dicionário inicial é

$$\begin{array}{rlllll} z = & 0 & +2x_1 & +x_2 & 0x_3 & \\ x_5 = & 5 & -x_1 & & -x_3 & \\ x_6 = & 10 & & -x_2 & & -x_4 \\ x_7 = & 0 & +x_1 & & & -x_4 \\ x_8 = & 0 & -x_1 & & & +x_4 \end{array}$$

Após o pivô x_1–x_8 obtemos

$$\begin{array}{rlllll} z = & 0 & -2x_8 & +x_2 & & +2x_4 \\ x_5 = & 5 & +x_8 & & -x_3 & -x_4 \\ x_6 = & 10 & & -x_2 & & -x_4 \\ x_7 = & 0 & -x_8 & & & \\ x_1 = & 0 & -x_8 & & & +x_4 \end{array}$$

Após o pivô x_4–x_5 obtemos

$$\begin{array}{rrrrrr} z = & 10 & & +x_2 & -2x_3 & -2x_5 \\ x_4 = & 5 & +x_8 & & -x_3 & -x_5 \\ x_6 = & 5 & -x_8 & -x_2 & +x_3 & +x_5 \\ x_7 = & 0 & -x_8 & & & \\ x_1 = & 5 & & & -x_3 & -x_5 \end{array}$$

Após o pivô x_2–x_6 obtemos a solução final

$$\begin{array}{rrrrrr} z = & 15 & -x_8 & -x_6 & -x_3 & -x_5 \\ x_4 = & 5 & +x_8 & & -x_3 & -x_5 \\ x_2 = & 5 & -x_8 & -x_6 & +x_3 & +x_5 \\ x_7 = & 0 & -x_8 & & & \\ x_1 = & 5 & & & -x_3 & -x_5 \end{array}$$

Questão 0.5 (Programação inteira)

1. Sim. Um programa linear pode ser resolvido em tempo polinomial, enquanto o problema da programação inteira é NP-complete.
2. Sim. Sem variáveis inteiras, um programa inteiro é um programa linear. Portanto o número de variáveis inteiras em geral determina mais a complexidade da solução do problema que o número de restrições.
3. Não. A solução relaxada arredondada nem necessáriamente é uma solução válida.
4. Não, é o contrário, ela é um super-conjunto, porque a relaxação é menos restrita, e portanto possui mais soluções.
5. Sim. Como a região viável do sistema inteira é menor, uma solução da relaxação linear é um limite superior para solução inteira. Caso essa solução ainda é inteira, então ele tem que ser ótima.