

Para cada um dos seguintes problemas formule como programa inteiro (puro, 0-1, ou misto), implementa o modelo em Julia/JuMP e resolve o modelo. Você pode aplicar as técnicas para formular condições lógicas e custos fixos visto em aula.

Laboratório 2

Questão 1 (Formulação)

Uma empresa foi contratada para cinco projetos, mas por problemas internas só pode concluir três. Para cada projeto não concluído ela tem que pagar uma multa. Cada projeto executado tem custos iniciais e um lucro proporcional à número de unidades produzidas. O número de unidades produzidas é decisão da empresa, até um limite de demanda. O total das unidades produzidas em todos projetos não pode ser mais que 10000. A seguinte tabela mostra os valor das multas, custos iniciais, lucros por unidade e demandas para cada projeto:

Projeto	1	2	3	4	5
Multa m_i [KR\$]	17	8	5	8	8
Custo inicial c_i [KR\$]	5	5	7	7	8
Lucro/unidade l_i [R\$]	10	11	13	17	11
Demanda [1000 unid] d_i	2	8	3	1	20

Os projetos 1 e 2 são do mesmo cliente e só podem ser executados ambos ou nenhum. O projeto 5 só pode ser executado, se o projeto 4 for executado também. Formule um programa inteiro que maximiza o lucro da empresa.

Questão 2 (Eleições)

Suponha que uma unidade federativa (UF) manda D pessoas à assembléia legislativa do Brasil. Existem M municípios na UF ($M > D$) e a UF quer agrupar esses municípios em D distritos eleitorais, tal que cada distrito manda uma pessoa à assembléia. A população total da UF é P , e o objetivo é que cada distrito eleitoral possui uma população de aproximadamente $p = P/D$. Suponha que a comissão que define os distritos gerou uma lista longa de N *candidatos* para distritos ($N > D$). Cada distrito candidato consiste em algumas municípios (inteiras) com população total p_j ($1 \leq j \leq N$), que é suficientemente perto de p . Seja $c_j = |p_j - p|$. Cada município faz parte de ao menos um distrito candidato, e tipicamente faz parte de um número considerável de distritos candidatos (para permitir várias seleções alternativas de D distritos candidatos tal que cada município é incluído exatamente uma vez). Seja

$$a_{ij} = \begin{cases} 1, & \text{se município } i \text{ faz parte do distrito candidato } j, \\ 0, & \text{caso contrário.} \end{cases}$$

Dado c_j e a_{ij} , o objetivo é selecionar D dos N possíveis distritos eleitorais, tal que cada município faz parte de exatamente um distrito, e tal que a *maior* das c_j correspondentes é o menor possível.

Formule um programa inteiro que resolve esse problema.

Questão 3 (Formulação matemática)

Queremos otimizar o acesso a um banco de dados. Tem cinco buscas que queremos aplicar. Além disso tem 5 opções de calcular índices que aceleram as buscas. A seguinte tabela mostra o custo da cada busca (em segundos) em função do índice

Busca	Índice					
	0	1	2	3	4	5
1	59	41	59	26	53	58
2	93	93	23	84	62	27
3	79	38	32	79	52	02
4	97	41	97	16	69	39
5	97	75	10	58	20	97

A tabela também contém um “índice 0” que é custo base de cada busca caso nenhum dos outros índices está disponível. Além disso tem um custo fixo para criar cada dos índices 1 a 5 e um consumo de memória, como segue:

	Índices				
	1	2	3	4	5
Custo fixo	86	28	03	48	25
Memória	34	21	17	56	79

O objetivo é minimizar o custo total, que consiste do custo fixo para preparar os índices mais o custo das buscas. O uso total de memória não pode ser mais que $D = 100$.

Questão 4 (Formulação, Takuzu)

Um *Takuzu* é uma grade 8×8 com dicas que precisa ser preenchida completamente com 0s e 1s, tal que tem o mesmo número de 0s e 1s em cada linha e coluna e tem no máximo dois 0s ou 1s adjacentes.

			0	0			
1							
						1	
1			0				0
		1					
0							
0			0	0			0
					1	1	

Formula um programa inteiro que resolve Takuzus, maximizando os 1s.