Problema PL/4

{Bronson [1982], Cap. 6, *Integer programming: branch-and-bound algorithm*, Problema 6.13 (p 62)}

Resolva o *Problema 1.20*.

{Bronson [1982], Cap. 1, *Mathematical programming*[a], Problema 1.20 (p 15)}

"Problema da mochila" (**KP,** "knapsack problem")

Uma câmara municipal tem 250 k$ (10^3 $[b]) orçamentados para a instalação de novas áreas de tratamento de lixo. Estão disponíveis 7 localizações, cujas capacidades projectadas (tonelada por semana, t[c] / semana) e custos de implantação (k$) são dados na Tabela adiante.

Que localizações deve a câmara seleccionar ?

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Total
Localização	*A*	*B*	*C*	*D*	*E*	*F*	*G*	
Capacidade, t/sem.	20	17	15	15	10	8	5	90
Custo, k$	145	92	70	70	84	14	47	522

■ Resolução

Equacionamento do problema

O objectivo é, claramente, maximizar a capacidade total instalada, atendendo aos constrangimentos existentes, o mais evidente dos quais[d] é, no caso presente, *de natureza orçamental*. A variável de decisão será um vector (de dimensão 7) contendo, como elementos, variáveis que devem ser *binárias* —é o outro constrangimento—, cada uma das quais indicará se na correspondente localização haverá implantação ou não.

O problema em estudo é análogo ao *problema da mochila*, ou *"knapsack problem"*, clássico. Neste, um campista ou montanhista necessita de colocar na sua mochila um máximo de objectos "valiosos" (latas de mantimentos, instrumentos, etc.), não podendo cada um deles ser fraccionado, mas o peso total está limitado. Por exemplo, usando a tabela dada: haveria 7 objectos úteis, cujos pesos se leriam na linha "custo" e não poderiam ultrapassar 250 (mas, ao todo, com peso de 522); e o valor (subjectivo) dos objectos estaria na linha "capacidade". Então, *F* seria o objecto mais valioso, com uma razão benefício sobre custo de 8/14 = 0,57, máxima.

Embora haja vários algoritmos específicos, mais eficientes, para o "knapsack problem", resolvê-lo-emos recorrendo à Programação Linear, concretamente, Programação Linear Inteira, ou Binária.

Modelo matemático

Verifica-se adiante que o modelo matemático do problema, exceptuando a última condição (variáveis binárias 0–1), é constituído exclusivamente por equações

[a] Definir-se-á "mathematical programming", programação matemática.

[b] Unidade monetária arbitrária (dólar, euro, etc.).

[c] "t", tonelada, *ton* ou [Thompson & Taylor, 2008, p 8] *metric ton* (British, *tonne*), errado na Table 6 (*op. cit.*), onde figura como T.

[d] Outros constrangimentos serão, precisamente, os que caracterizam o KP.

lineares. Esta característica torna o problema um caso particular da Programação Matemática, que é a *Programação Linear*. Além disso, a introdução do constrangimento suplementar de que algumas variáveis (ou todas) têm de ser inteiras transforma-o num caso de *Programação Linear Inteira Mista*, MILP ("mixed integer linear programming"); e, ainda, o facto de as variáveis inteiras serem binárias leva-o à designação de BILP, ou BIP ["binary (mixed) integer linear programming"].

[max] $z = 20\,x_1 + 17\,x_2 + 15\,x_3 + 15\,x_4 + 10\,x_5 + 8\,x_6 + 5\,x_7$

sujeito a: $145\,x_1 + 92\,x_2 + 70\,x_3 + 70\,x_4 + 84\,x_5 + 14\,x_6 + 47\,x_7 \leq 250$

com: $x_i \leq 1 \qquad i = 1..7$

$(x_i \geq 0)$

Todas as variáveis **inteiras**

O grupo de constrangimentos $x_i \geq 0$ —os chamados "constrangimentos implícitos"— foi colocado entre parêntesis não porque não seja fundamental, mas para lembrar que é respeitado automaticamente pelo algoritmo.

Os três últimos grupos de constrangimentos podem também ser expressos na forma

$$x_i = 0; 1 \qquad i = 1..7$$

ou ainda, no contexto da Programação Linear, dizendo simplesmente

Todas as variáveis **binárias**

(embora o termo "binário" possa, em rigor, referir-se a dois valores não forçosamente 0 e 1).

Procedimento via "branch-and-bound"

Combinar-se-á a Programação Linear com o algoritmo de "branch-and-bound" (B&B) —"ramificação e limite", ou "bifurcação e limite", mas poucas vezes traduzido, ou, mais consagrado, "partição e avaliação sucessivas" [Guerreiro *et al.*, 1985, *op. cit.*]—, da autoria de Land e Doig [1960], que pertence à classe dos chamados *métodos de enumeração*.

A resolução inicia-se com o problema original *excluídos os constrangimentos de integralidade* ("relaxação"), constituindo o que podemos chamar o "Problema 1". A solução deste por Programação Linear mostra-se na Tabela 1.

Tabela 1

Probl.	z	Esg$^{\underline{do}}$.*	x_1	x_2	x_3	x_4	x_5	x_6	x_7	De; para
1	$55{,}55^+$	—	$0{,}027^+$	1	1	1	0	1	0	—

**Esgotado*: (ainda não aplicável) estado de "esgotamento" do *problema*.

Ora, a solução do Problema 1 ***não é admissível***, visto que não satisfaz os constrangimentos de integralidade: x_1 não é inteiro. Entretanto, ficou conhecido um *limite* (máximo) absoluto para z, que é 55,55, visto que, como recorreremos à

introdução de *mais constrangimentos*, o óptimo final só pode, evidentemente, "piorar" (exactamente, não melhorar).

NOTA A propósito de algarismos significativos, bastaria escrever $z = 55{,}5$, ou mesmo 55^{+}: acontece que, neste problema, os *coeficientes* (na função objectivo) têm máximo divisor comum (*mdc*) 1, portanto, qualquer resultado admissível será múltiplo de 1 (pois que as variáveis virão a ser *todas* inteiras). O óptimo do problema original, logo, será, na melhor das hipóteses, o valor (múltiplo do *mdc*) 55.

O Problema 1 vai ramificar-se em dois novos problemas (**partição** do Problema 1):

– O Problema 2, proveniente do constrangimento adicional $x_2 \leq 0$; e

– O Problema 3, proveniente do constrangimento adicional $x_2 \geq 1$.

Certamente, notamos que $x_2 \leq 0$ implica $x_2 = 0$, o que, se substituído no Problema 2, o facilitaria —e sem dúvida assim faríamos em cálculo manual—; porém, não recorreremos a particularidades numéricas neste exemplo-tipo.

Os resultados dos Problemas 2 e 3 são agora introduzidos na Tabela 2.

Tabela 2

Probl.	z	Esg$^{\underline{do}}$.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	De; para
1	55,55	—	0,02	1	1	1	0	1	0	–; 2, 3
2	55,47	—	$\boxed{0}$	1	1	1	0,04	1	0	1; 4, 5
3	47,5		$\boxed{1}$	0	1	0,3	0	1	0	

O Problema 1 deixou de interessar: diremos que está *esgotado* após "sondagem" ("fathomed"[e]). No entanto, nem o Problema 2 nem o Problema 3 satisfazem os constrangimentos de integralidade, havendo, pois, que continuar. A **avaliação** das soluções, z, presentes, dos Problemas 2 e 3 (respectivamente, 55,47 e 47,5), leva-nos a escolher a primeira, por ser mais promissora (maior). Assim, o Problema 2 dará origem aos dois novos Problemas 4 e 5, conforme x_5 fique abaixo ou acima de 0,04, i. é, $x_5 \leq 0$ ($x_5 = 0$) e $x_5 \geq 1$ ($x_5 = 1$), e ficará esgotado.

Havendo mais do que uma variável não-inteira, sugere-se optar por aquela que esteja mais longe de ser inteira, i. é, com parte decimal *mais próxima de 0,5* (e desempata-se arbitrariamente)[f]**.**

Tabela 3

Probl.	z	Esg$^{\underline{do}}$.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	De; para
1	55,55	—	0,02	1	1	1	0	1	0	–; 2, 3
2	55,47	—	$\boxed{0}$	1	1	1	0,04	1	0	1; 4, 5
3	47,5		$\boxed{1}$	0	1	0,3	0	1	0	

[e] *Fathom* [´fæð ə m], toeza (medida de 6 pés) usada na medição náutica da profundidade; *to fathom*, profundar, medir profundidade, sondar. (De *further arm*, [medida dos] braços estendidos.)

[f] E ainda (porquê ?), se fossem variáveis *inteiras* (não só *binárias*), *v.g.*, 7,5 seria "menos inteiro" que 8,5.

4	55,42		[0]	1	1	1	[0]	1	0,08	
5	50,21		[0]	0,13	1	1	[1]	1	0	

Continua a não haver solução binária. Das soluções disponíveis, Problemas 3, 4 e 5, a mais promissora é a do Problema 4; deste, originar-se-ão os Problemas 6 e 7, bifurcando em x_7.

Tabela 4

Probl.	z	Esg$^{\underline{do}}$.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	De; para
1	55,55	—	0,02	1	1	1	0	1	0	–; 2, 3
2	55,47	—	[0]	1	1	1	0,04	1	0	1; 4, 5
3	47,5		[1]	0	1	0,3	0	1	0	
4	55,42	—	[0]	1	1	1	[0]	1	0,08	2; 6, 7
5	50,21		[0]	0,13	1	1	[1]	1	0	
6	55	Corr* **Opt**	[0]	1	1	1	[0]	1	[0]	
7	52,05		[0]	0,53	1	1	[0]	1	[1]	

* Corr: solução corrente.

Obtivemos uma solução "binária", i. é, que satisfaz todos os constrangimentos do problema original. Esta solução fica esgotada e é a ***solução corrente***, de valor $z = 55$. Então —uma vez que as restantes soluções disponíveis não podem melhorar este valor (nem sequer igualá-lo, o que poderia dar soluções múltiplas)—, a solução corrente é **óptima** e a resolução do problema está terminada:

$$\mathbf{x} = [\ 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0\ ]^T \qquad z = 55$$

Ou seja, serão seleccionadas 4 localizações —2ª., 3ª., 4ª. e 6ª., designadas ***B***, ***C***, ***D*** e ***F***—, com uma capacidade total de **55** tonelada / semana (e um custo total que se calcula em 246 k$, face ao disponível de 250 mil).

Uma representação gráfica progressiva da "árvore" das soluções do problema tem bastante interesse (Figura 1).

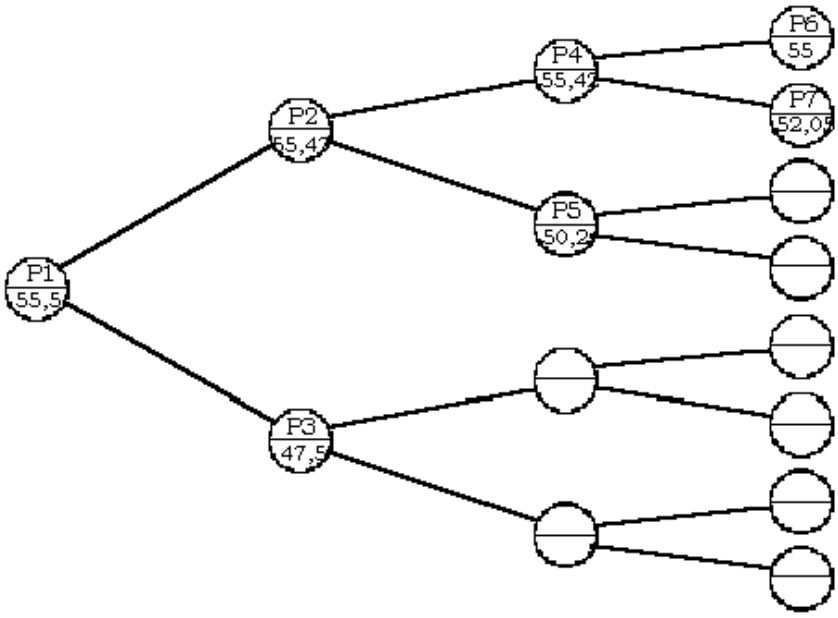

Figura 1

Programação Matemática e Programação Linear

Um problema de Programação Matemática pode ser definido como

$$\begin{aligned} [\text{opt}] \quad & z = f(\mathbf{x}) \\ \text{sujeito a} \quad & \mathbf{g}(\mathbf{x}) \,\underline{r}\, \mathbf{0} \\ \text{com} \quad & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

com
z –valor da *função objectivo*
f –função objectivo a optimizar (i. é, minimizar ou maximizar)
$\mathbf{x}$ –vector das variáveis independentes
$\mathbf{g}$ –vector das funções de constrangimento
$\underline{r}$ –conjunto ordenado de relações ≤, = e ou ≥

Um problema de Programação Linear é um caso particular, no qual tanto a função f como as funções $\mathbf{g}$ são lineares.

Referências

– BRONSON, Richard, 1982, "Theory and Problems of Operations Research", Schaum's Outline Series, McGraw-Hill Book Company, New York, NY (USA).

– LAND, A. H. and A. G. DOIG, 1960, "An automatic method for solving discrete programming problems", *Econometrica*, **28** (3): 497–520.

– TAYLOR, Ambler, and Barry N. TAYLOR, 2008, "Guide for the use of the International System of Units (SI)", NIST Special Publication 811, National Institute for Standards and Technology, Gaithersburg, MA (USA).

■

➢ **KNAPSACK**

• **knap·sack** (näp´säk´) —***n.*** A bag, as of canvas or nylon, worn on the back to carry supplies and equipment. [LG *knappsack* : *knappen*, to bite + *sack*, bag.] [HERITAGE, 1986]

O Heritage usa "ä" em vez de "æ" —e muitas outras notações, sempre empíricas— e não os símbolos do Alfabeto Fonético Internacional ("Associação Fonética Universal"). Coloca um *acento agudo* ***após*** a sílaba "tónica", e outros (se houver subtónicas —sendo então o acento principal grafado em normando). Os "pontos" intercalados nas palavras indicam locais para translineação.

• **knap** (næp)

knap[1] —*tr. v.* **knapped**, **knap·ping**, **knaps** **1.** CHIEFLY BRITISH. REGIONAL. To strike sharply; rap. **2.** To break or chip flints with a sharp blow. **3.** CHIEFLY BRITISH. REGIONAL. To snap at or bite. [ME *knappen*] **knap´per** *n.*

knap[2] —*n.* REGIONAL. The crest of a hill; summit. [ME < OE *cnæp*]

Porém, o Oxford não regista **knap** ! [HERITAGE, 1986]

• **knap** [næp], (1) *vt.* KNAPPING [-iŋ], KNAPPED [-t], partir, quebrar; estalar; britar. — (2) *s.* eminência; elevação.

• **knapsak** [´næpsæk], *s.* **MOCHILA** de soldado; saco de campismo. [Ferreira, 1954]

Provavelmente, gralha, por **knapsack**.

• **sack** (sæk)

sack[1] —*n.* **1. a.** A large bag of strong, coarse material for holding objects in bulk. **b.** A similar container of paper or plastic. **c.** The amount held in a sack. **2.** Also **sacque.** A short, loose-fitting garment for women and children. **3.** SLANG. A dismissal from employment: *finally got the sack.* **4.** SLANG. A bed, mattress, or sleeping bag. **5.** A baseball base.

—*tr. v.* **sacked**, **sack·ing**, **sacks** **1.** To place in a sack. **2.** SLANG. To discharge from employment. —**Phrasal verb: sack out.** SLANG. To sleep. [ME < OE *sæcc* < Lat. *saccus* < Gk. *sakkos*, of Semitic orig.]

sack[2] —*tr. v.* **sacked**, **sack·ing**, **sacks** To rob of goods or valuables, esp. after capture.

—*n.* **1.** The looting or pillaging of a captured town. **2.** Plunder; loot. [< OFr. *(mettre a) sac, (to put in) a sack.*]

• **sac** (sæk) —***n.*** A pouch or pouchlike structure in a plant or animal, sometimes filled with fluid. [Fr., bag < Lat. *saccus*. —see sack[1].] [HERITAGE, 1986]

• **sac** [sæk], *s.* saco, bolsa, cavidade.

• **sack** [sæk], *s.* (1) saco, saca, taleigo, costal; saque, pilhagem (mil.); … — (2) *vt.* ensacar, saquear; despedir. [Ferreira, 1954]

Referências

– FERREIRA, Júlio Albino, "Dicionário Inglês-Português", Nova edição, Editorial Domingos Barreira, Porto, 1954 (data do prefácio do revisor)

– [HERITAGE] "The American Heritage Dictionary", Houghton Mifflin Company, 1987, parte da "*Microsoft Bookshelf* CD-ROM Reference Library", Multimedia 1993 Edition

– [OXFORD] FOWLER, Francis George and Henry Watson FOWLER, "The Pocket Oxford Dictionary of current English", 7th. ed. (1984, 1924), Oxford University Press, Oxford OX2 6DP, Great Britain, reprinted 1985

Problema PL/5

Resolva o seguinte problema como Programação Linear Binária.

$$[\max] \qquad z = \quad 20\,x_1 + 17\,x_2 + 15\,x_3$$

$$\text{sujeito a:} \qquad 145\,x_1 + 92\,x_2 + 70\,x_3 \leq 250$$

■ **Resolução**

Suporemos *(para evitar os cálculos dos vários problemas de Programação Linear pelo leitor)* que dispomos, antecipadamente, das soluções dos seguintes 27 problemas, sendo o Problema 0 a programação linear original relaxada dos constrangimentos de integralidade e os restantes Problemas (1 a 26) formados pelo Problema 0 acompanhado dos constrangimentos em x_1, x_2 e x_3 adiante indicados.

Tabela 5

	Constrangimentos			Solução			
Probl.	x_1	x_2	x_3	z	x_1	x_2	x_3
0	—	—	—	44,14	0,6	1	1
1	—	—	0	37	1	1	0
2	—	—	1	44,14	0,6	1	1
3	—	0	—	35	1	0	1
4	—	0	0	20	1	0	0
5	—	0	1	35	1	0	1
6	—	1	—	44,14	0,6	1	1
7	—	1	0	37	1	1	0
8	—	1	1	44,14	0,6	1	1
9	0	—	—	32	0	1	1
10	0	—	0	17	0	1	0
11	0	—	1	32	0	1	1
12	0	0	—	15	0	0	1
13	0	0	0	0	0	0	0
14	0	0	1	15	0	0	1
15	0	1	—	32	0	1	1
16	0	1	0	17	0	1	0
17	0	1	1	*imp.*	—	—	—
18	1	—	—	41,47	1	0,4	1
19	1	—	0	37	1	1	0
20	1	—	1	41,47	1	0,4	1
21	1	0	—	35	1	0	1
22	1	0	0	20	1	0	0
23	1	0	1	35	1	0	1
24	1	1	—	39,79	1	1	0,2
25	1	1	0	37	1	1	0
26	1	1	1	*imp.*	—	—	—

imp.: problema impossível

Com base nesta informação, podemos construir a Tabela 6, que contém o percurso da obtenção da solução óptima.

Tabela 6

Probl.	z	Esg^{do}.	x_1	x_2	x_3	De; para	
0	44,14	—	0,6	1	1	–; 9, 18	
9	32	Corr—	[0]	1	1	0	
18	41,47	—	[1]	0,4	1	0; 21, 24	
21	35	Corr—	[1]	[0]	1	18	
24	39,79	—	[1]	[1]	0,2	18; 25, 26	
25	37	Corr	[1]	[1]	[0]	24	**opt**
26	*imp.*	—	[1]	[1]	[1]	24	

A solução é, assim:

$$\mathbf{x} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T \qquad z = 37.$$

Na Figura 2, mostra-se a árvore de soluções em forma gráfica. Na realidade, só seriam calculados os Problemas 0, 9 & 18, 21 & 24 e 25 & 26 (possivelmente, apelidados de Problemas 1, 2, 3, etc.) e não os 27 problemas anteriormente tabelados — vantagem do "branch-and-bound".

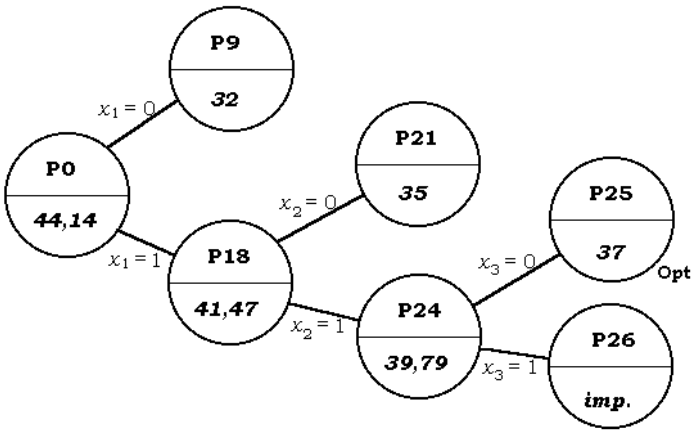

Figura 2

Os Problemas 1, 7, e 19, que, pela Tabela 5, parecem soluções alternativas, são, de facto, idênticos à solução óptima (i. é, são a *mesma* solução —mesmo **x**); surgiram devido ao modo artificioso por que a resolução foi apresentada. A solução do problema é, pois, única. (Note-se que a natureza do algoritmo leva a que, caso haja soluções múltiplas, possam surgir durante o mero processo de cálculo.)

■

❖