Enumeração Implícita Aplicada à Seleção de Investimentos

*Samsão Woiler**

1. Modelos de Programação Matemática. 2. Algoritmo.

A aplicação da programação matemática teve um grande impulso com um trabalho apresentado por WEINGARTNER.[1] Entretanto, por não existirem algoritmos eficientes para resolver problemas de programação linear com variáveis inteiras, o citado trabalho preocupou-se principalmente em desenvolver modelos de programação linear para a seleção de investimentos.

WEINGARTNER[2] definiu diversos tipos de propostas de investimento: investimentos mùtuamente exclusivos, aquêles dos quais se pode selecionar no máximo um; investimentos dependentes, que só podem ser aprovados se os investimentos dos quais êles dependem forem aprovados; e investimentos independentes, que não estão ligados a nenhuma outra proposta. Investimentos mùtuamente exclusivos e investimentos dependentes podem ser considerados depois de resolvido o problema de programação linear, mas nos modelos apresentados neste artigo êles serão considerados na formulação inicial. Quando não consideramos a interdependência entre os projetos e empregamos a programação linear, podemos ter dificuldades com as respostas fracionais para investimentos independentes que sejam fundamentais

* Livre docente da Escola Politécnica da Universidade de São Paulo.
1 WEINGARTNER, H. M. *Mathematical Programming and the Analysis of Capital Budgeting Problems*, Englewood Cliffs, Prentice-Hall Inc., 1963.
2 *Idem, ibidem.*

para a aceitação de investimentos dependentes. Um teorema apresentado por WEINGARTNER[3] mostra que o número máximo de projetos com respostas fracionais numa solução por programação linear é o número de períodos com restrições orçamentárias mais o número de restrições expressando relações de interdependência.

1. Modelos de Programação Matemática

Nesta seção apresentamos um modelo matemático de programação linear com variáveis inteiras bivalentes para resolver o problema da seleção de investimentos; os investimentos que forem aceitos estarão associados à variáveis com valor unitário e os outros a variáveis com valor 0.

No que segue, usaremos a seguinte notação:

a) c_{tk} é o valor recebido (>0) ou pago (<0) no fim de período t se aceitarmos a proposta k de investimento.

b) x_k é a variável associada à proposta k. Valerá 1 se a proposta fôr aceita e zero no caso contrário.

c) K é o número de propostas que serão analisadas. Evidentemente $k = 1, 2, \ldots, K$.

d) n é o número de períodos que utilizaremos na análise dos investimentos. É o horizonte fixado no caso de seleção de investimentos futuros. Portanto, $t = 0, 1, 2, \ldots, n$.

e) b_t são fundos provenientes de fontes diferentes das propostas consideradas e que se encontram disponíveis no fim do período t.

Êles podem provir de investimentos já feitos antes da data de início da seleção em questão ou de outra fonte qualquer, independente das propostas em estudo.

f) c_{nk} é o valor residual da proposta k no fim do n-ésimo período. Ou seja, é o valor presente dos valôres a serem recebidos e pagos após o término do período $(n-1)$ computada a taxa de retôrno da proposta k. Assumiremos em nosso modêlo que cada proposta tenha uma única taxa de retôrno.

[3] *Idem, ibidem.*

g) r é a taxa de juros por período paga pelos bancos.

h) V_t é o saldo no fim do período t.

Inicialmente, apresentaremos um modêlo de programação linear inteira, assumindo que as propostas sejam independentes. Em seguida, mostraremos como acrescentar as restrições de interdependência entre os projetos.

De acôrdo com a notação considerada anteriormente, podemos descrever o problema da seleção de investimentos com o seguinte modêlo matemático:

$$\text{maximizar } V_n,\; V_n = b_n + (1 + r)\, V_{n-1} + \sum_{k=1}^{K} c_{nk}\, x_k$$

sujeito a

$$-\sum_{k=1}^{K} c_{0k}\, x_k \leq b_0, \text{ no início do 1.º período}$$

$$-\sum_{k=1}^{K} c_{1k}\, x_k \leq b_1 + (1 + r)\, V_0, \text{ no fim do 1.º período}$$

$$-\sum_{k=1}^{K} c_{(n-1)k}\, x_k \leq b_{n-1} + (1 + r)\, V_{n-2}, \text{ no fim do período } n - 1$$

onde,

$$x_k = 0 \text{ ou } 1,$$

$$V_0 = b_0 + \sum_{k=1}^{K} c_{0k}\, x_k \geq 0,$$

$$V_1 = b_1 + \sum_{k=1}^{k} c_{1k}\, x_k + (1 + r)\, V_0 \geq 0$$

$$V_{n-1} = b_{n-1} + \sum_{k=1}^{K} c_{(n-1)k}\, x_k + (1 + r)\, V_{n-2} \geq 0.$$

O modêlo anterior é muito complexo, e sabendo que r é muito pequeno podemos assumir que os juros pagos pelos bancos não são investidos nas propostas consideradas. Considerando-se a hipótese acima, o modêlo pode ser expresso da seguinte forma:

$$\text{maximizar } V_n = \sum_{t=0}^{n} b_t + \sum_{t=0}^{n} \sum_{k=1}^{K} c_{tk}\, x_k,$$

sujeito a

$$- \sum_{k=1}^{K} c_{0k}\, x_k \leq b_o, \text{ no início do 1.º período}$$

$$- \sum_{k=1}^{K} c_{1k}\, x_k \leq b_o + \sum_{k=1}^{K} c_{0k}\, x_k + b_1, \text{ no fim do 1.º período}$$

e

$$- \sum_{k=1}^{K} c_{(n-1)k}\, x_k \leq \sum_{t=0}^{n-1} b_t + \sum_{t=0}^{n-2} \sum_{k=1}^{K} c_{(n-2)k}\, x_k \text{ , no fim do}$$

período $(n-1)$

onde,

$$x_k = 0 \text{ ou } 1.$$

Além de ser mais simples, o segundo modêlo está formulado apenas em função das variáveis x_k. Entretanto, a segunda formulação tem mais têrmos em cada restrição.

Como os valôres b_t são parâmetros independentes das variáveis x_k, podemos simplificar a função-objetivo para

$$\text{maximizar} \sum_{t=0}^{n} \sum_{k=1}^{K} c_{tk}\, x_k.$$

Se as propostas $k_1, k_2, \ldots, k_n$ forem mùtuamente exclusivas, deveremos acrescentar uma restrição do tipo

$$\sum_{j=1}^{m} x_{k_j} \leq 1,$$

que garantirá que no máximo uma das propostas seja aceita.

Se a aceitação da proposta k_d depender da aceitação da proposta k_i, acrescentaremos uma restrição do tipo $x_{k_d} - x_{k_i} \leq 0$, que impedirá que x_{k_d} possa ser igual a 1 com x_{k_i} igual a zero.

Os símbolos b_t e c_{kt} são parâmetros do modêlo e, portanto, deverão ser substituídos antes de aplicarmos os algoritmos para resolver o problema.

O número de períodos (n) e o número de propostas (K) que podem ser examinados pelo modêlo dependem do programa e do computador que se devem usar.

Segundo WEINGARTNER,[4] o problema da seleção de investimentos foi colocado pela primeira vez na forma de um modêlo de programação linear inteira por J. H. LORIE e L. J. SAVAGE. O modêlo básico por êles desenvolvido foi o seguinte:

$$\text{maximizar} \sum_{k=1}^{K} c_k \, x_k ,$$

sujeito a

$$\sum_{k=1}^{K} a_{tk} \, x_k \leq b_t , \quad t = 0, 1, \ldots, T$$

onde $x_k = 0$ ou 1 para $k = 1, 2, \ldots, K$.

A diferença fundamental entre o modêlo acima e os anteriores está na utilização dos coeficientes c_k, que podem ser substituídos pelos valôres presentes dos investimentos a uma dada taxa de juros real, se considerarmos os valôres a_{tk} deflacionados a preços do instante em que $t = 0$.

O modêlo de LORIE e SAVAGE pode ser aplicado a diversos casos práticos, mesmo que introduzamos a hipótese de que todos os parâmetros a_{kt} sejam substituídos por valôres positivos. Por exemplo, PETERSON considerou essa hipótese na seleção de propostas de pesquisa e desenvolvimento. No algoritmo descrito a seguir, consideramos essa mesma hipótese[5].

2. Algoritmo

Antes de definirmos a notação a ser usada na descrição do algoritmo de enumeração implícita por nós introduzido, recordaremos alguns conceitos encontrados na literatura.

Uma *solução parcial* fica definida quando se especificam valôres para as variáveis de um subconjunto das n variáveis originais. As variáveis para as quais não forem especificados valôres serão chamadas *livres*. Cada vez que especificarmos um valor para uma das variáveis livres, *aumentaremos* a solução parcial. Se especificarmos valôres para tôdas

[4] *Idem, ibidem.*

[5] PETERSEN, C. C. Computational Experience with Variants of the Balas Algorithm Applied to the Selection of R. and D. Projects, *in Management Science,* vol. 13, n.º 9, 1967, p. 736 a 750.

as variáveis livres de uma solução parcial, definiremos uma descendente da mesma.

Nos algoritmos de *enumeração implícita* procuramos sempre manter um conjunto de soluções parciais cujas descendentes incluam no mínimo uma solução ótima, e através de modificações e testes de eliminação tentamos reduzir êste conjunto até que seus elementos sejam apenas soluções ótimas.

2.1. NOTAÇÃO

O algoritmo descrito nesta seção utiliza o clássico *backtracking* de GLOVER[6], GEOFFRION[7] e outros. Êste método de enumeração exige apenas um vetor de n componentes, $u = (u_1, u_2, \ldots, u_n)$, para especificar o estado da enumeração implícita.

Sejam x_{j_1} $x_{j_2}, \ldots, x_{j_k}$ as k variáveis com valôres especificados na solução parcial s_q obtida na q-ésima iteração. Faremos $u_k = j_k$ se $x_{j_k} = 1$ e seu complemento ainda não foi considerado e $u_k = -j_k$ se $x_{j_k} = 0$ ou 1 e seus respectivos complementos já foram considerados no processo de enumeração implícita. Quando os testes de eliminação mostram que s_q não tem descendente viável, o elemento positivo mais à direita do vetor u é multiplicado por -1, e todos os componentes negativos à sua direita são anulados. A enumeração terminará quando todos os componentes forem não-positivos, desde que inicialmente $u_1, u_2, \ldots, u_n$ sejam iguais a zero.

Seja $(s_q, 0)$ a solução obtida quando se consideram as variáveis $x_{j_1}, x_{j_2}, \ldots, x_{j_k}$ iguais aos valôres especificados e as variáveis livres iguais a zero.

Antes de apresentarmos o diagrama de bloco e os testes usados na eliminação de soluções parciais, introduziremos a seguinte notação:

$z = \sum_{k=1}^{K} c_{j_k} x_{j_k}$ é o valor da função-objetivo para a solução $(s_q, 0)$. Desde que $c_j \geq 0$ seja o mínimo valor que poderemos obter, se considerarmos tôdas as descendentes de s_q.

6 GLOVER, F. A Multiphase Dual Algorithm for the Zero-One Integer Programming Problem, *in Operations Research,* vol. 13, n.º 6, 1965, p. 879 a 919.

7 GEOFFRION, A. M. *Integer Programming by Implicit Enumeration and Balas Method,* Califórnia, The Rand Corporation, RM-4783 — PR, 1966.

$$\hat{z} = \sum_{j=1}^{n} c_j \hat{x}_j$$

é uma limitação superior para o valor ótimo da função-objetivo, onde $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$ é a melhor solução viável obtida até a q-ésima iteração.

$$y_i = -b_i + \sum_{k=1}^{K} a_{ij_k} x_{j_k} \geq 0$$

determina se a restrição i está satisfeita ou não; de fato, se algum y_i fôr negativo, a solução $(s_q, 0)$ não é viável; entretanto, s_q pode ter descendentes viáveis.

$$d_i = \sum_{k=K+1}^{n} (+\ a_{ij_k})$$

é a soma de todos coeficientes positivos, tais que x_{j_k} é uma variável livre.

$$d_i^* = \sum_{f \in T_i} a_{i1}$$

onde T_i é o conjunto de todos os índices f, tais que x_f é livre, $z + c_f < \hat{z}$ e $a_{if} > 0$.

$$w_i = y_i + d_i \quad \text{e} \quad w_i^* = y_i + d_i^*$$

determinam se a restrição i pode ser satisfeita ou não.

Seja i_o um particular índice i, tal que $y_{i_o} < 0$, e j_o um particular índice j, tal que x_{j_o} é variável livre.

2.2. DIAGRAMA DE BLOCO

No diagrama de bloco que segue o modêlo de LORIE e SAVAGE será considerado na seguinte forma:

$$\text{minimizar} \sum_{j=1}^{n} c_j x_j$$

sujeito a

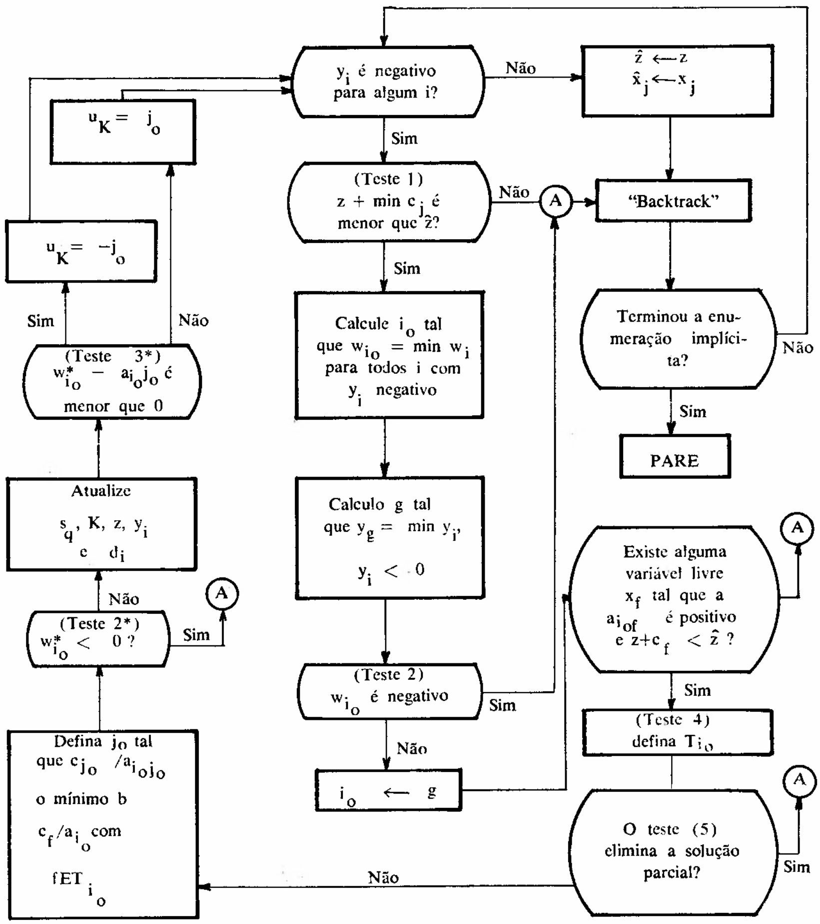
$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i,\ i = 1, 2, \ldots, m$$

onde $x_j = 0$ ou 1, $c_j \geq 0$, $a_{ij} \geq 0$ e b_i é irrestrito.

Para passarmos da forma inicial para esta última, temos que considerar:

$$n = K$$
$$x_j = x_k, \text{ para } j = k,\ k = 1, 2, \ldots, K$$
$$m = T + 1$$
$$a_{ij} = a_{tk}, \text{ para } i = t + 1, t = 0, 1, \ldots, T, \text{ e } j = k, k = 1, 2, \ldots, K$$
$$b_i = \sum_{j=1}^{n} a_{ij} - b_t \text{ para } t = i - 1$$

FIGURA 1: *Esquema do Algoritmo para Problemas com Coeficientes Positivos*



2.3. TESTES

Baseados na notação anterior, os seguintes testes foram aplicados para a eliminação da solução parcial s_q:

1. Se $z + \min_{j=1}^{n} c_j \geq \hat{z}$, podemos eliminar s_q, pois não existem descendentes dessa solução parcial que melhorem a solução viável conhecida, $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$.

2. Se $w_{i_o} < 0$, podemos eliminar s_q, pois não existem descendentes viáveis dessa solução parcial.

3. Se $w_{i_o} > 0$ e $w_{i_o} - a_{i_o j_o} < 0$, então devemos ter necessàriamente $x_{j_o} = 1$.

4. Se T_{i_o} é vazio, podemos eliminar s_q.

5. Consideremos o seguinte problema:
   minimizar $\sum_f c_f \, x_f$,
   sujeito a
   $y_{i_o} + \sum_f a_{if} \, x_f \geq 0$, $f E \, T_{i_o}$ e
   $x_f = 0$ ou 1.

Se $x_{f_1}, x_{f_2}, \ldots, x_{f_{r-1}}, x_{f_r}$ é uma solução ótima do problema, onde apenas x_{f_r} pode ser fração, então

$$L_{i_o} = z + c_{f_1} + c_{f_2} + \ldots + c_{f_{r-1}}$$

é uma limitação inferior para o valor da função-objetivo da melhor descendente da solução parcial s_q. Portanto, se $L_{i_o} \geq \hat{z}$, podemos eliminar s_q, pois não poderemos obter nenhuma solução melhor que $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n$, se continuarmos usando os valôres fixados para $x_{j_1}, x_{j_2}, \ldots, x_{j_k}$.

Testes (2*) e (3*), obtidos quando substituímos w_i por w_i^* nos testes (2) e (3), foram introduzidos por BALAS[8].

2.4. RESULTADOS

Utilizando um programa em Algol no Computador Bourroughs 5500 da Universidade de Stanford, resolvemos o problema 5 de PETERSON[9],

[8] BALAS, E. An Additive Algorithm for Solving Linear Programs with Zero — One Variables, *in Operations Research*, vol. 13, n.º 4, 1965, p. 517 a 546.
[9] PETERSEN, C. C. Computational Experience with Variants of the Balas Algorithm Applied to the Selection of R. and D. Projects, *in Management Science*, vol. 13, n.º 9, 1967, p. 736 a 750.

com 10 restrições e 28 variáveis, em 20,2 segundos. Êsse tempo de processamento é menor do que todos os tempos já observados na solução do referido problema conforme podemos observar na tabela 1.

TABELA 1: *Tempos para o problema Petersen n.º 5*

Algoritmos	Tempo	Fonte
LAR - FIR LAR-FIR-MUL	20,9 seg 21,5 seg	Woiler (1967)
Reform. Dalas	46 min	Petersen (1967)
Modif. G-1	20 min	
Modif. G-1 e G-2	29 min	
Modif. R-1	3 min	
Modif. R-2	10 min	
Modif. G-1 e R-2	3 min	

A melhora de 3,5% em relação ao melhor tempo por nós obtido anteriormente pode ser muito importante para problemas de maior dimensão. De fato, o tempo de computador deve ser bem aproveitado, devido ao seu custo elevado.

É importante lembrarmos que os algoritmos que se aplicam a problemas específicos sempre podem resolver com maior eficiência os problemas aos quais se aplicam. O algoritmo em questão é um exemplo dêsse fato.

Algoritmos para problemas mais gerais de seleção de investimentos poderão ser encontrados nos trabalhos por nós apresentados em 1967 e 1968.[10]

Bibliografia

BALAS, E. An Additive Algorithm for Solving Linear Programs with Zero — One Variables, *in Operations Research*, vol. 13, n.º 4, 1965, p. 517 a 546.

[10] WOILER, S. *Implicit Enumeration Algorithms for Discrete Optimization Problems*, Department of Industrial Engineering Tech. Report Stanford University, 1967; *Programação Linear com Variáveis Inteiras*: *Algoritmos e Aplicações*, tese de Livre Docência, Escola Politécnica, Universidade de São Paulo, 1968.

GEOFFRION, A. M. *Integer Programming by Implicit Enumeration and Balas Method*, Califórnia, The Rand Corporation, RM-4783 — PR, 1966.

GLOVER, F. A. Multiphase Dual Algorithm for the Zero — One Integer Programming Problem, *in Operations Research*, vol. 13, n.º 6, 1965, p. 879 a 919.

PETERSEN, C. C. Computational Experience with Variants of the Balas Algorithm Applied to the Selection of R. and D. Projects, *in Management Science*, vol. 13, n.º 9, 1967, p. 736 a 750.

WEINGARTNER, H. M. *Mathematical Programming and the Analysis of Capital Budgeting Problems*, Englewood Cliffs, Prentice-Hall Inc., 1963.

WOILER, S. *Implicit Enumeration Algorithms for Discrete Optimization Problems*, Department of Industrial Engineering Tech. Report Stanford University, 1967.

WOILER, S. *Programação Linear com Variáveis Inteiras: Algoritmos e Aplicações*, tese de Livre Docência, Escola Politécnica, Universidade de São Paulo, 1968.