Programação Linear: método Simplex

A programação linear apareceu na Matemática durante a II Grande Guerra. O objetivo era deslocar exércitos, armas, combustíveis, comida, e toda a logística associada. O matemático George Dantzig que era funcionário do Escritório de Controle Estatístico da Força Aérea Americana criou uma maneira inteligente de resolver este tipo de problema usando o algoritmo simplex. Note-se que a disponibilidade de computadores logo a seguir aumentou o interesse da coisa. Dantzig morreu em 2005 e o campo que ele inaugurou ainda está sob constante desenvolvimento.

Método Simplex

Seja o caso: Um fazendeiro pode criar bois ou ovelhas para lã. Um boi rende 8000 por ano, e uma tonelada de lã rende 21.000/ano. Para criar 1 boi usa-se 11 ha de área. Para 1 ton de lã usa-se 60 ha de área. Um boi consome 1h/dia de manuseio e para gerar 1 ton de lã usa-se 2h/dia de manuseio. Considerando que o fazendeiro tem 500 ha de área e trabalham ele e dois filhos (o que dá 24 h/dia). Achar o melhor plano. Definindo as variáveis: x_1 =bois; x_2 =toneladas de lã; x_3 =área não utilizada (folga) e x_4 =horas não utilizadas (folga também). Z é a lucratividade do negócio.

Tableaux

$Z - 8x_1 - 21x_2 = 0$
$11x_1 + 60x_2 + x_3 = 500$
$x_1 + 2x_2 + x_4 = 24$

Z	x_1	x_2	x_3	x_4	D
1	-8	-21	0	0	0
0	11	60	1	0	500
0	1	2	0	1	24

A coluna que deve entrar é a que tem o maior valor negativo na primeira linha, portanto a coluna x_2.

Z	x_1	x_2	x_3	x_4	D
1	-8	-21	0	0	0
0	11	60	1	0	500
0	1	2	0	1	24
		↑			

Para determinar a linha que vai sair da base, Divida o lado direito (D) pelo coeficiente da coluna pivot (só se este for estritamente positivo (> 0) e identifique a linha que possui a menor razão: $500/60 = 8.33$ e $24/2 = 12$, a linha menor é a linha do 500, e fica

Z	x_1	x_2	x_3	x_4	D	
1	-8	-21	0	0	0	
0	11	60	1	0	500	←
0	1	2	0	1	24	
		↑				

Agora temos a coluna pivot e a linha pivot e no cruzamento de ambos, o pivot.

Chegou o momento de operações elementares com linhas.

* Divida a linha pivot pelo pivot. Esta é a NOVA linha pivot. Fica

Z	x_1	x_2	x_3	x_4	D	
1	-8	-21	0	0	0	
0	0.183	1	0.016	0	8.33	←
0	1	2	0	1	24	
		↑				

* Para cada outra linha (inclusive a primeira)

- Se o coeficiente da coluna pivot é negativo: $L_{atual} = L_{atual} + |coeficiente| \times nova - linha - pivot$

- se positivo: $L_{atual} = L_{atual} - |coeficiente| \times nova - linha - pivot$

Primeira linha: $1, -8, -21, 000 + |21| * 0, 0.183, 1, 0.016, 0, 8.33$ e fica: $1, -4.157, 0, 0.336, 0, 174.93$
Terceira linha: $0, 1, 2, 0, 1, 24 - |2| * 0, 0.183, 1, 0.016, 0, 8.33$ e fica: $0, 0.634, 0, -0.032, 1, 7.34$

Z	x_1	x_2	x_3	x_4	D
1	-4.157	0	-0.032	1	7.34
0	0.183	1	0.016	0	8.33
0	0.634	0	-0.032	1	7.34
	↑				

Linha pivot: candidatos: a segunda 8.33/0.183 ou a terceira 7.34/0.634 é a terceira.
A nova linha pivot é: $0, 1, 0, -0.0504, 1.577, 11.57$
A nova linha 1 é: $1, 0, 0, -0.241, 7.555, 55.436$
A nova linha 2 é: $0, 0, 1, 0.0252, -0.288, 6.212$ e fica

Z	x_1	x_2	x_3	x_4	D
1	0	0	-0.2415	7.555	55.436
0	1	0	-0.0504	1.577	11.57
0	0	1	0.025	-0.288	6.212

Para finalizar, a resposta é $x_1 = 11.57$ e $x_2 = 6.21$. O valor da função objetivo é $L = 8 * x_1 + 21 * x_2 = 8 * 11.57 + 21 * 6.21 = 223.00$ (arredondando)

```
from pulp import *
import numpy as np
import matplotlib.pyplot as plt
prob=LpProblem("fazenda",LpMaximize)
X1=LpVariable('X1',0,None,None)
X2=LpVariable('X2',0,None,None)
prob+=8*X1+21*X2,"Lucro total"
prob+=11*X1+60*X2<=500,"restricao de area"
prob+=X1+2*X2<=24,"restricao de horas"
prob.writeLP("c:/p/python/lp03.lp")
prob.solve()
print("Status ",LpStatus[prob.status])
for v in prob.variables():
  print(v.name,' = ',v.varValue)
print("Total ",value(prob.objective))
Status  Optimal
X1  =  11.578947   X2  =  6.2105263
Total  223.05262829999998
```

O problema

Começa com uma variável (custo, lucro, vazão, produção ou qualquer outra coisa similar) que se quer maximizar ou minimizar. Continua com uma lista de restrições que são inequações que o modelo deve obedecer. Tais desigualdades devem ser lineares e elas modelam o ambiente onde o problema se insere. O objetivo da programação linear é mostrar o valor otimizado e sobretudo as condições que vão permitir alcançá-lo.

Um exemplo

Existe uma fábrica de 4 produtos: A,B,C e D. Ela deseja maximizar o lucro de seus quatro produtos A, B, C e D. Para fabricá-los usam-se as máquinas F=Furadeira e P=Prensa e dois tipos de mão de obra O=Operário e S=Supervisor. As disponibilidades mensais das máquinas são $F = 80$, $P = 20$. Quanto aos trabalhadores a disponibilidade é $O = 120$ e $S = 160$ horas por mês. As necessidades de produção por milhar de produto são:

Insumo	A	B	C	D
Furadeira	5	4	8	9
Prensa	2	6	-	8
Operário	2	4	2	8
Supervisor	7	3	-	7

O setor comercial da empresa informa que os potenciais de venda e os lucros unitários de cada produto são

Produto	Potencial vendas	L. unitário
argolas	70	10
botões	60	8
colchetes	40	9
decotes	20	7

O problema pede as quantidades que deverão ser produzidas dos quatro produtos a fim de que o lucro seja maximizado. Chamemo-las de A, B, C, D. Da tabela de lucro, obtemos que $L = 10 \times A + 8 \times B + 9 \times C + 7 \times D$.

As restrições do problema são:
$A \leq 70$; $B \leq 60$; $C \leq 40$; $D \leq 20$ e mais
$5A + 4B + 8C + 9D \leq 80$
$2A + 6B + 8D \leq 20$
$2A + 4B + 2C + 8D \leq 120$
$7A + 3B + 7D \leq 160$,
e $A, B, C, D \geq 0$.

Usando o algoritmo simplex, (por exemplo através do programa LP88), obtem-se como resposta: $A = 10$; $B = 0$; $C = 3,75$; $D = 0$; com lucro de $L = 133,75$.

Eis a programação em Python

```
from pulp import *
prob=LpProblem("folha",LpMaximize)
A=LpVariable('A',0,None,None)
B=LpVariable('B',0,None,None)
C=LpVariable('C',0,None,None)
D=LpVariable('D',0,None,None)
prob+=10*A+8*B+9*C+7*D,"Lucro total"
prob+=A<=70,"potencial de vendas de A"
prob+=B<=60,"potencial de vendas de B"
prob+=C<=40,"potencial de vendas de C"
prob+=D<=20,"potencial de vendas de D"
prob+=5*A+4*B+8*C+9*D<=80,'restricao furadeira'
prob+=2*A+6*B+8*D<=20,'restricao prensa'
prob+=2*A+4*B+2*C+8*D<=80,'restricao operario'
prob+=7*A+3*B+7*D<=160,'restricao supervisao'
prob.writeLP("c:/p/python/lp02.lp")
prob.solve()
print("Status ",LpStatus[prob.status])
for v in prob.variables():
  print(v.name,' = ',v.varValue)
print("Total ",value(prob.objective))

>>> Status  Optimal
A=10.0   B=0.0   C=3.75   D=0.0    Total=133.75
```

Eis a matriz entregue ao programa simplex caseiro para este problema

```
-10 -8 -9 -7 0 0 0 0 0 0 0 0   0
  5  4  8  9 0 0 0 0 1 0 0 0  80
  2  6  0  8 0 0 0 0 0 1 0 0  20
  2  4  2  8 0 0 0 0 0 0 1 0 120
  7  3  7  0 0 0 0 0 0 0 0 1 160
  1  0  0  0 1 0 0 0 0 0 0 0  70
  0  1  0  0 0 1 0 0 0 0 0 0  60
  0  0  1  0 0 0 1 0 0 0 0 0  40
  0  0  0  1 0 0 0 1 0 0 0 0  20
```

☞ Para você fazer

Procure em um livro de Pesquisa Operacional (como por exemplo Introdução à Pesquisa Operacional de HILLIER, Frederick e LIEBERMAN, Gerald, ou Programação Linear de Edgard Lanzer, ou qualquer outro, um problema qualquer de PL. Transcreva-o, arme as variáveis e as equações e depois resolva-o usando o pacote PULP do Python. Entregue tudo junto com esta folha.