Programação Linear - método simplex: situações particulares

Investigação Operacional

J.M. Valério de Carvalho
vc@dps.uminho.pt

Departamento de Produção e Sistemas
Escola de Engenharia, Universidade do Minho

21 de outubro de 2020

Prog. Linear - método simplex: situações particulares

antes

- O método Simplex foi aplicado para resolver um problema de programação linear.

Guião

- Há situações particulares em que é necessário detalhar as regras e estabelecer decisões e operações suplementares:
    - quando o domínio é ilimitado;
    - quando não existe um vértice inicial admissível.
    - quando há vértices degenerados;

depois

- Analisaremos a implementação do método simplex usando matrizes.

Situações particulares do método simplex:

- Selecção de um vértice admissível inicial
  - Se não existir, **problema é impossível**.
- Repetir
  - Selecção da coluna pivô:
    - Coeficiente mais negativo na linha da função objectivo
    - (em caso de empate, escolha arbitrária)
    - Se não existir coef.<0, solução óptima.
  - Selecção da linha pivô:
    - Menor razão (lado direito/coluna pivô) positiva (*i.e.*, coef.col.>0)
    - (em caso de empate, o próximo vértice é **degenerado)**
    - Se não existir coef.col.>0, **solução óptima é ilimitada**.
  - Fazer eliminação de Gauss
- Enquanto (solução não for óptima)

Teorema (Fundamental de Programação Linear)

Dado um problema de programação linear, se não existir uma solução óptima com valor finito, então ou o problema é impossível ou a solução óptima é ilimitada.

O que vimos e o que vamos ver prova o Teorema.

Conteúdo

1. Domínio ilimitado (aberto)

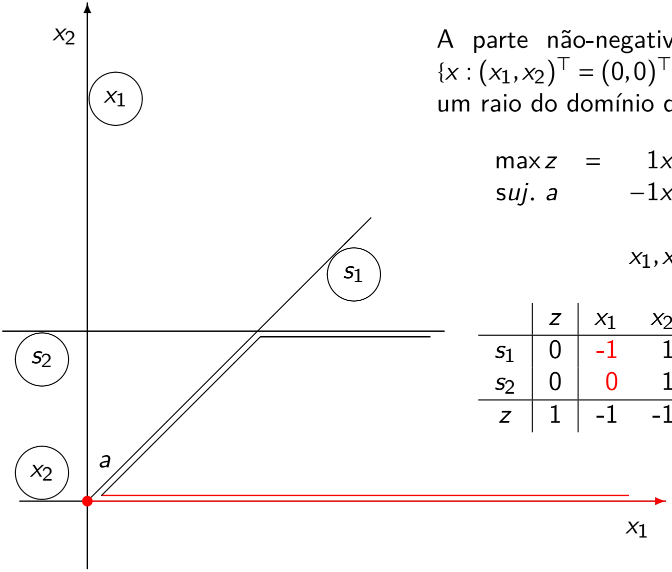
Definição:

- O domínio de um problema de programação linear é um poliedro ilimitado (aberto) se contiver um raio ($\equiv$ semi-recta).

Um raio é caracterizado por um ponto e uma direcção:

- O raio R é o conjunto de pontos:
  - com início no ponto $v \in \mathbb{R}^n$ e
  - ao longo da direcção $d \in \mathbb{R}^n$ (um vector não-nulo), *i.e.*,
- $R = \{ x : x = v + \theta . d, \theta \geq 0 \}$.

O domínio de um problema de programação linear é um poliedro fechado se e só se não contiver nenhum raio, podendo então ser representado como uma combinação convexa dos seus vértices.



A parte não-negativa do eixo de x_1, $\{x : (x_1,x_2)^\top = (0,0)^\top + \theta(1,0)^\top, \theta \geq 0\}$, é um raio do domínio do problema:

$$\begin{array}{llrrl} \max z & = & 1x_1 & +1x_2 & \\ suj.\ a & & -1x_1 & +1x_2 & \leq 0 \\ & & & +1x_2 & \leq 2 \\ & & x_1, x_2 \geq 0 & & \end{array}$$

	z	x_1	x_2	s_1	s_2	
s_1	0	-1	1	1	0	0
s_2	0	0	1	0	1	2
z	1	-1	-1	0	0	0

Domínio ilimitado: como identificar no quadro simplex?

Quadro simplex: como identificar um raio?

- Há uma coluna de uma variável não-básica em que os coeficientes das restrições são todos ≤ 0 (no exemplo, os elementos a vermelho).

- Exemplo:

$$\left\{ \begin{array}{lllr} s_1 & = & 0+1x_1 & -1x_2 \\ s_2 & = & 2 & -1x_2 \\ z & = & 0+1x_1 & +1x_2 \end{array} \right.$$

	z	x_1	x_2	s_1	s_2	
s_1	0	-1	1	1	0	0
s_2	0	0	1	0	1	2
z	1	-1	-1	0	0	0

Ao longo de um raio, todos os pontos são admissíveis, porque:

- há uma **única** variável não-básica que aumenta de valor,
- todas as vars básicas aumentam (quando o coef.<0) ou mantêm o valor (quando o coef.$=0$),
- sendo portanto todas as suas coordenadas ≥ 0.

Domínio ilimitado: solução óptima ilimitada

- A solução óptima de um problema é ilimitada quando, ao longo de um raio, o valor da função objectivo melhora.

Quadro simplex: como identificar uma solução óptima ilimitada?

- há um raio e
- o respectivo coeficiente na linha da função objectivo é:
  - < 0 (em problemas de maximização), ou
  - > 0 (em problemas de minimização).

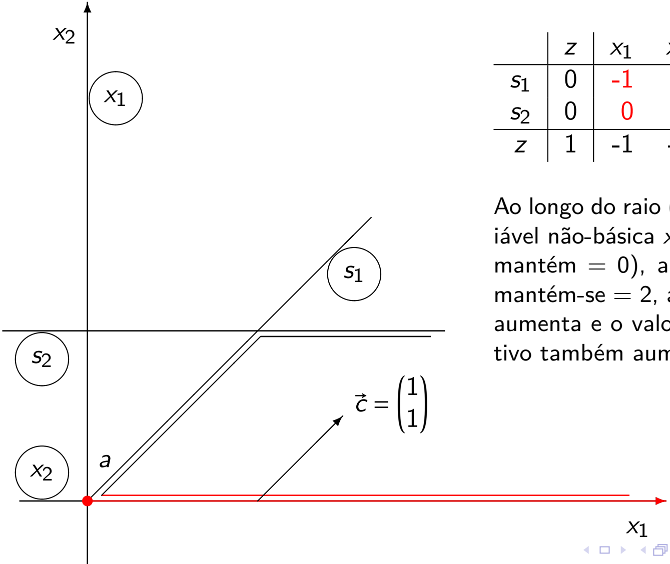
- Exemplo (problema de maximização):

$$\begin{cases} s_1 = 0 + 1x_1 - 1x_2 \\ s_2 = 2 - 1x_2 \\ z = 0 + 1x_1 + 1x_2 \end{cases}$$

	z	x_1	x_2	s_1	s_2	
s_1	0	-1	1	1	0	0
s_2	0	0	1	0	1	2
z	1	-1	-1	0	0	0

O corpo central do quadro simplex (as restrições) dá informação sobre o domínio. A linha da função objectivo do quadro simplex dá informação sobre o valor da solução óptima, que pode ser finito ou ilimitado.

Exemplo: domínio ilimitado e solução óptima ilimitada



	z	x_1	x_2	s_1	s_2	
s_1	0	-1	1	1	0	0
s_2	0	0	1	0	1	2
z	1	-1	-1	0	0	0

Ao longo do raio (*i.e.*, quando a variável não-básica x_1 aumenta e x_2 se mantém $= 0$), a variável básica s_2 mantém-se $= 2$, a variável básica s_1 aumenta e o valor da função objectivo também aumenta.

2. Vértice admissível inicial

- E se não estiver imediatamente disponível um vértice admissível (quadro simplex) inicial?
- Para ilustrar essa situação, vamos usar um exemplo em que há restrições de $\geq$.
- É necessário usar o Método das 2 Fases
    - Fase I: obter um vértice admissível inicial
    - Fase II: aplicar algoritmo simplex

Um problema com restrições de $\geq$ e de minimização

- No seguinte problema, não é possível identificar imediatamente um vértice admissível inicial:

$$
\begin{array}{lrcrcrcr}
\min z = & 120y_3 & + & 80y_4 & + & 30y_5 & & \\
& 3y_3 & + & 1y_4 & + & 1y_5 & \geq & 12 \\
& 2y_3 & + & 2y_4 & & & \geq & 10 \\
& \multicolumn{7}{l}{y_3, y_4, y_5 \geq 0}
\end{array}
$$

Transformação na forma canónica

$$
\begin{array}{lcl}
\min z & = & cx \\
& & Ax \geq b \\
& & x \geq 0
\end{array}
\quad \rightarrow \quad
\begin{array}{lcl}
\min z & = & cx \\
& & Ax - u = b \\
& & x, u \geq 0
\end{array}
$$

sendo $u \in \mathbb{R}_+^{m\times 1}$ um vector de variáveis de folga da mesma dimensão que $b \in \mathbb{R}^{m\times 1}$.

Transformação Inequações → Equações

- Qualquer inequação do tipo $\geq$ pode ser transformada numa equação (equivalente), introduzindo uma variável adicional, designada por *variável de folga*, com valor não-negativo.
- Exemplo:

$$\left\{ \begin{array}{ll} 3x_1 + 2x_2 & \geq 120 \\ x_1, x_2 & \geq 0 \end{array} \right. \Rightarrow \left\{ \begin{array}{ll} 3x_1 + 2x_2 - 1u_1 & = 120 \\ x_1, x_2, u_1 & \geq 0 \end{array} \right.$$

- O número de unidades produzidas numa solução $(x_1, x_2)^\top$ é igual ao valor da função linear: $3x_1 + 2x_2$.
- u_1 (variável de folga) é o número de unidades produzidas em excesso relativamente às necessárias (no exemplo, 120).
- Há autores que designam estas variáveis por *variáveis de excesso*.

Exemplo: transformação na forma canónica

Modelo original

- Variáveis de decisão: y_3, y_4, y_5.

$$
\begin{array}{rrcrcrcr}
\min z = & 120y_3 & + & 80y_4 & + & 30y_5 & & \\
 & 3y_3 & + & 1y_4 & + & 1y_5 & \geq & 12 \\
 & 2y_3 & + & 2y_4 & & & \geq & 10 \\
 & \multicolumn{7}{l}{y_3, y_4, y_5 \geq 0}
\end{array}
$$

Modelo na forma canónica (equivalente ao modelo original)

- Variáveis de decisão: y_3, y_4, y_5.
- Variáveis de excesso: y_1, y_2.

$$
\begin{array}{rccrcrcrl}
\min z = & & & 120y_3 & + & 80y_4 & + & 30y_5 & \\
 & -1y_1 & + & 3y_3 & + & 1y_4 & + & 1y_5 & = 12 \\
 & -1y_2 & + & 2y_3 & + & 2y_4 & & & = 10 \\
 & \multicolumn{8}{l}{y_1, y_2, y_3, y_4, y_5 \geq 0}
\end{array}
$$

Não há um vértice admissível inicial, porque ...

- o lado direito é positivo e não há uma matriz identidade no quadro:

	z	y_1	y_2	y_3	y_4	y_5	
	0	-1	0	3	1	1	12
	0	0	-1	2	2	0	10

Lembrete: antes havia um vértice admissível inicial, porque:

- as restrições eram todas de $\leq$ (havia uma matriz identidade $I_{m \times m}$), e
- os coeficientes do lado direito eram todos ≥ 0.

- Quando não há um vértice admissível inicial, usa-se o:

Método das 2 Fases:

- na Fase I, resolve-se um *problema auxiliar* para tentar encontrar um vértice admissível inicial.
- Se se conseguir, na Fase II, aplica-se o algoritmo simplex; caso contrário, o problema é impossível.

Método das 2 fases: estratégia

Fase I: adicionar variáveis artificiais e minimizar a sua soma

- resolver problema auxiliar ($\mathbf{1}a$ é a soma das variáveis artificiais):

$$\begin{aligned} \min z_a &= \mathbf{1}a \\ & Ax - u + a = b \\ & x, u, a \geq 0 \end{aligned}$$

  sendo $a \in \mathrm{I\!R}_+^{m\times 1}, \mathbf{1} = [1, 1, \ldots, 1]$ um vector linha com m elementos.
- Se $(\min z_a = \mathbf{1}a = 0) \Rightarrow a = \tilde{0}$ (todas as variáveis artificiais $= 0$) $\Rightarrow$ há um vértice admissível que obedece às restrições originais;
- caso contrário $(\min z_a > 0)$, não é possível obter uma solução que obedeça a todas as restrições originais $\Rightarrow$ problema é impossível.

Fase II: optimizar problema original

- Existe um vértice admissível inicial para o algoritmo simplex;
- optimiza-se a função objectivo (original) do problema.

Fase I: adicionar vars artificiais a_1 e a_2, e $\min z_a$

- Função objectivo da Fase I: $\min z_a \;=\; 1a_1 + 1a_2.$
- Equação da linha da função objectivo: $z_a - 1a_1 - 1a_2 = 0$

minimizar a sua soma

variáveis artificiais

	z_a	y_1	y_2	y_3	y_4	y_5	a_1	a_2	
a_1	0	-1	0	3	1	1	1	0	12
a_2	0	0	-1	2	2	0	0	1	10
z_a	1	0	0	0	0	0	- 1	- 1	0

objetivo → $a_1 = 0 \wedge a_2 = 0$

- O quadro não é válido: os coeficientes da linha da função objectivo debaixo da matriz identidade devem ser nulos. ▸ Validar Quadro

- O quadro seguinte é válido; vamos ▸ minimizar a função auxiliar z_a:

	z_a	y_1	y_2	y_3	y_4	y_5	a_1	a_2	
a_1	0	-1	0	$\boxed{3}$	1	1	1	0	12
a_2	0	0	-1	2	2	0	0	1	10
z_a	1	-1	-1	$\boxed{5}$	3	1	0	0	22

maior

não ótimo

válido

neste método, como estamos a minimizar a escolha da coluna é diferente
↳ escolher o mais positivo

Fase I: iterações

	z_a	y_1	y_2	y_3	y_4	y_5	a_1	a_2	
a_1	0	-1	0	$\boxed{3}$	1	1	1	0	12
a_2	0	0	-1	2	2	0	0	1	10
z_a	1	-1	-1	5	3	1	0	0	22

	z_a	y_1	y_2	y_3	y_4	y_5	a_1	a_2	
y_3	0	-1/3	0	1	1/3	1/3	1/3	0	4
a_2	0	2/3	-1	0	$\boxed{4/3}$	-2/3	-2/3	1	2
z_a	1	2/3	-1	0	4/3	-2/3	-5/3	0	2

	z_a	y_1	y_2	y_3	y_4	y_5	a_1	a_2	
y_3	0	-1/2	1/4	1	0	1/2	1/2	-1/4	7/2
y_4	0	1/2	-3/4	0	1	-1/2	-1/2	3/4	3/2
z_a	1	0	0	0	0	0	-1	-1	0

- Solução óptima: $\min z_a = 0$.
- Foi encontrado um vértice admissível.

Fase I: conclusão

- O vértice admissível é $(y_1, y_2, y_3, y_4, y_5)^\top = (0, 0, 7/2, 3/2, 0)^\top$.
- Variáveis artificiais (a_1, a_2) e função objectivo auxiliar (z_a) não são necessárias na Fase II, e podem ser eliminadas.

		y_1	y_2	y_3	y_4	y_5	
y_3		-1/2	1/4	1	0	1/2	7/2
y_4		1/2	-3/4	0	1	-1/2	3/2

- Na Fase II, iremos optimizar a função objectivo original (z), partindo do vértice admissível encontrado na Fase I.

Fase II: função objectivo original

- Função objectivo da Fase II: $\min\ z = 120y_3 + 80y_4 + 30y_5$.
- Equação da linha da função objectivo: $z - 120y_3 - 80y_4 - 30y_5 = 0$

	z	y_1	y_2	y_3	y_4	y_5	
y_3	0	-1/2	1/4	1	0	1/2	7/2
y_4	0	1/2	-3/4	0	1	-1/2	3/2
z	1	0	0	-120	-80	-30	0

- O quadro não é válido: os coeficientes da linha da função objectivo debaixo da matriz identidade devem ser nulos.

Validar Quadro

- O quadro seguinte é válido; vamos optimizar a função original z:

	z	y_1	y_2	y_3	y_4	y_5	
y_3	0	-1/2	1/4	1	0	1/2	7/2
y_4	0	1/2	-3/4	0	1	-1/2	3/2
z	1	-20	-30	0	0	-10	540

- O primeiro vértice admissível encontrado é a solução óptima
- (problema de minimização e nenhum coeficiente na linha da função objectivo é positivo).
- Isto nem sempre acontece!

3. Vértices degenerados

- O que é um vértice degenerado?
- Como identificar a degenerescência no quadro simplex?

Vértice degenerado: caracterização

- Vimos vértices determinados pela intersecção de $(n - m)$ hiperplanos.

Vértice degenerado: número maior de hiperplanos

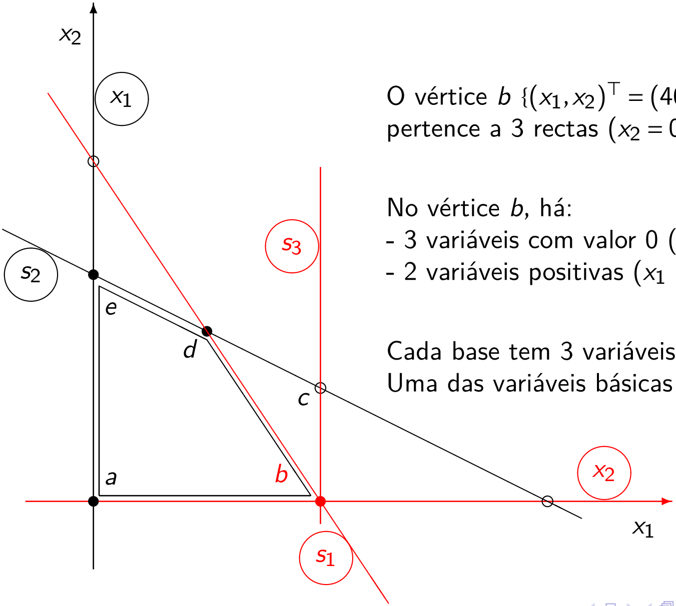
- Um vértice *degenerado* pertence a mais do que $(n - m)$ hiperplanos.

- Lembrete: uma solução básica resulta de resolver o sistema de m equações em ordem a m variáveis básicas, associadas a um conjunto de m vectores linearmente independentes, que formam a *base*.

Vértice degenerado: várias bases, a mesma solução básica ($\equiv$ vértice)

- Um vértice é *degenerado* se várias bases (cada uma correspondendo a um quadro simplex diferente) fornecerem a mesma solução básica.

Exemplo: 3 hiperplanos no espaço a 2 dimensões



O vértice b $\{(x_1, x_2)^\top = (40, 0)^\top\}$ é degenerado; pertence a 3 rectas ($x_2 = 0, s_1 = 0$ e $s_3 = 0$).

No vértice b, há:

- 3 variáveis com valor 0 (x_2, s_1 e s_3) e
- 2 variáveis positivas (x_1 e s_2).

Cada base tem 3 variáveis básicas.
Uma das variáveis básicas é nula.

Ver mais

Exemplo: 3 bases diferentes, a mesma solução básica

	z	x_1	x_2	s_1	s_2	s_3	
s_1	0	0	2	1	0	-3	0
s_2	0	0	2	0	1	-1	40
x_1	0	1	0	0	0	1	40
z	1	0	-10	0	0	12	480

	z	x_1	x_2	s_1	s_2	s_3	
x_2	0	0	1	0.5	0	-1.5	0
s_2	0	0	0	-1	1	2	40
x_1	0	1	0	0	0	1	40
z	1	0	0	5	0	-3	480

	z	x_1	x_2	s_1	s_2	s_3	
s_2	0	0	4/3	-1/3	1	0	40
s_3	0	0	-2/3	-1/3	0	1	0
x_1	0	1	2/3	1/3	0	0	40
z	1	0	-2	4	0	0	480

Um quadro simplex corresponde a um *vértice degenerado* se houver uma ou mais variáveis básicas com valor 0.

Solução básica ($\equiv$ vértice) é sempre $(x_1, x_2, s_1, s_2, s_3)^\top = (40, 0, 0, 40, 0)^\top$.

Escolha da linha pivô quando há empate

	z	x_1	x_2	s_1	s_2	s_3		
s_1	0	$\boxed{3}$	2	1	0	0	120	$120/3 = 40$
s_2	0	1	2	0	1	0	80	$80/1 = 80$
s_3	0	1	0	0	0	1	40	$40/1 = 40$
z	1	-12	-10	0	0	0	0	

- Há empate na menor razão positiva $= 40$ (linhas de s_1 e de s_3).

Desempate usando a técnica de perturbação:

- perturbar o lado direito, adicionando ϵ, e
- calcular novamente a menor razão positiva.

- Exemplo:
  - Linha de $s_1 : (120 + \epsilon)/3 = 40 + \epsilon/3$
  - Linha de $s_3 : \ (40 + \epsilon)/1 = 40 + \epsilon$
- Linha pivô correcta: a de s_1 (menor razão positiva após perturbação)

Esta escolha tipicamente reduz o número de iterações.

Exemplo: resolução em que um vértice é degenerado

	z	x_1	x_2	s_1	s_2	s_3	
s_1	0	$\boxed{3}$	2	1	0	0	120
s_2	0	1	2	0	1	0	80
s_3	0	1	0	0	0	1	40
z	1	-12	-10	0	0	0	0

	z	x_1	x_2	s_1	s_2	s_3	
x_1	0	1	2/3	1/3	0	0	40
s_2	0	0	$\boxed{4/3}$	-1/3	1	0	40
s_3	0	0	-2/3	-1/3	0	1	0
z	1	0	-2	4	0	0	480

	z	x_1	x_2	s_1	s_2	s_3	
x_1	0	1	0	0.5	-0.5	0	20
x_2	0	0	1	-0.25	0.75	0	30
s_3	0	0	0	-0.5	0.5	1	20
z	1	0	0	3.5	1.5	0	540

Solução óptima $(x_1, x_2, s_1, s_2, s_3)^\top = (20, 30, 0, 0, 20)^\top$.

4. O algoritmo termina num número finito de passos?

Quando não há degenerescência, o algoritmo simplex converge para

- a solução óptima (finita) num número finito de pivôs, porque

4. O algoritmo termina num número finito de passos?

Quando não há degenerescência, o algoritmo simplex converge para

- a solução óptima (finita) num número finito de pivôs, porque o resultado da soma de um número infinito de melhorias do valor da função objectivo em cada pivô não pode ser um valor finito.

Quando há degenerescência, o algoritmo simplex pode entrar em ciclo,

- percorrendo ciclicamente as bases correspondentes ao mesmo vértice, quando se selecciona para coluna pivô a **coluna com o coeficiente mais negativo** na linha da função objectivo (prob. max.).

4. O algoritmo termina num número finito de passos?

Quando não há degenerescência, o algoritmo simplex converge para

- a solução óptima (finita) num número finito de pivôs, porque o resultado da soma de um número infinito de melhorias do valor da função objectivo em cada pivô não pode ser um valor finito.

Quando há degenerescência, o algoritmo simplex pode entrar em ciclo,

- percorrendo ciclicamente as bases correspondentes ao mesmo vértice, quando se selecciona para coluna pivô a **coluna com o coeficiente mais negativo** na linha da função objectivo (prob. max.).

No entanto, a finitude do algoritmo simplex é assegurada

- se se seleccionar para coluna pivô a **coluna com coeficiente negativo e menor índice**.

ver Bland, R. "New finite pivoting rules for the simplex method". Mathematics of Operations Research 2 (2): 103 - 107, 1977.

doi:10.1287/moor.2.2.103

Complexidade do algoritmo simplex

- Há um exemplo especialmente construído (um hipercubo deformado no espaço a $n-$dimensões), em que o algoritmo simplex percorre todos os vértices.
- No espaço a 3 dimensões, percorre os $2^3 = 8$ vértices do cubo.
- No pior caso, o algoritmo simplex é exponencial.
- Em termos de comportamento médio, há estudos computacionais de implementações do algoritmo simplex em que o número de iterações se aproxima bem de $(m+n)/2$.

ver Klee V, Minty GJ (1972) How good is the simplex algorithm? In: Shisha O. (ed.) Inequalities: III. Academic Press, New York

Vanderbei, Linear Programming: Foundations and Extensions, International Series in Operations Research & Management Science, 2014

Conclusão

- A regra de Bland assegura a convergência do algoritmo simplex num número finito de passos.
- Há outros algoritmos para resolver problemas de programação linear, como os métodos de pontos interiores (que são polinomiais).
- O algoritmo simplex permanece competitivo, embora tenham sido identificados exemplos em que os métodos de pontos interiores têm melhor desempenho.

Apêndices

A.1. Método do Grande M: estratégia

associar uma penalidade muito grande às vars artificiais, para forçar que tenham um valor nulo

- resolver problema auxiliar:

$$\begin{aligned} \min z_M &= cx + \mathbf{M}a \\ & Ax - u + a = b \\ & x, u, a \geq 0 \end{aligned}$$

  sendo $a \in \mathbb{R}_+^{m \times 1}, \mathbf{M} = [M, M, \ldots, M]$ um vector linha com m elementos.

- Se M for suficientemente grande, qualquer ponto admissível é melhor do que um ponto em que uma variável artificial seja positiva.
- Se $(a = \widetilde{0})$ (todas as variáveis artificiais $= 0$) $\Rightarrow \min z_M = cx^*$ e x^* é o vértice admissível óptimo, que obedece às restrições originais.
- caso contrário $(\exists a_i > 0)$, não é possível obter uma solução que obedeça a todas as restrições originais $\Rightarrow$ problema é impossível.

A.1. Método do Grande M: desvantagens

Se o valor de M for muito grande,

- pode haver perda de informação, resultante da representação dos números em computador.
- Os coeficientes de custo são representados por reais de dupla precisão com um número finito de casas decimais.
- Exemplo:

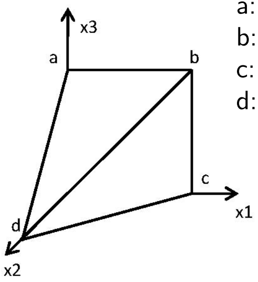
$$
\begin{aligned}
c_1 &= 3,1415926535897932e+00 \\
M &= 1,0000000000000000e+40 \\
M + c_1 &= 1,0000000000000000e+40
\end{aligned}
$$

Se o valor de M for muito pequeno,

- pode não ser suficientemente grande para conduzir todas as variáveis artificiais a 0.

A.2. Degenerescência e restrições redundantes

- Num vértice degenerado, pode haver uma *restrição redundante*, *i.e.*, uma restrição que pode ser removida sem alterar o domínio.
- Isso não acontece na generalidade. Há casos em que nenhuma restrição pode ser removida.
- Exemplo: as restrições que definem o vértice d : $x_1 + x_2 \leq 1$, $x_2 + x_3 \leq 1$ e $x_1, x_2, x_3 \geq 0$ são todas necessárias:



a: $x_2 = 0$; $x_1 = 0$; $x_2 + x_3 = 1$
b: $x_2 = 0$; $x_2 + x_3 = 1$; $x_1 + x_2 = 1$
c: $x_2 = 0$; $x_3 = 0$; $x_1 + x_2 = 1$
d: $x_3 = 0$; $x_1 = 0$; $x_1 + x_2 = 1$; $x_2 + x_3 = 1$(há 4 planos)

A.2. Degenescência e bases óptimas

	z'	x_1	x_2	s_1	s_2	s_3	
s_1	0	0	$\boxed{2}$	1	0	-3	0
s_2	0	0	2	0	1	-1	40
x_1	0	1	0	0	0	1	40
z'	1	0	-1	0	0	3	120

	z'	x_1	x_2	s_1	s_2	s_3	
x_2	0	0	1	0.5	0	-1.5	0
s_2	0	0	0	-1	1	2	40
x_1	0	1	0	0	0	1	40
z'	1	0	0	1/2	0	3/2	120

	z'	x_1	x_2	s_1	s_2	s_3	
s_2	0	0	4/3	-1/3	1	0	40
s_3	0	0	-2/3	-1/3	0	1	0
x_1	0	1	2/3	1/3	0	0	40
z'	1	0	1	1	0	0	120

Se $z' = 3x_1 + 1x_2$, uma das três bases da solução básica óptima não é óptima.

O quadro 1 é uma base que não é óptima: é necessário fazer um pivô degenerado para se comprovar que a solução básica (que não se altera quando se faz o pivô) é uma solução óptima.

Algoritmo simplex de minimização

Lembrete: no algoritmo simplex de minimização:

- a coluna pivô é a coluna com o coeficiente mais positivo na linha da função objectivo,
- a solução é óptima se não existir nenhum coeficiente positivo na linha da função objectivo.

NOTA: Em alternativa a usar um algoritmo de minimização, podemos usar um algoritmo simplex de maximização para maximizar a função simétrica da função objectivo. Ver diapositivos sobre Transformações básicas.

Voltar

I - Obter quadro válido: folha de rascunho

	z_a	y_1	y_2	y_3	y_4	y_5	a_1	a_2	
a_1	0	-1	0	3	1	1	1	0	12
a_2	0	0	-1	2	2	0	0	1	10
z_a	1	0	0	0	0	0	- 1	- 1	0

- Exprimir a função objectivo z_a em função das variáveis não-básicas y_1, y_2, y_3, y_4 e y_5 usando eliminação de Gauss: somar à linha de z_a as linhas de a_1 e a_2.

	z_a	y_1	y_2	y_3	y_4	y_5	a_1	a_2	
$(+1)$*linha de z_a	1	0	0	0	0	0	-1	-1	0
$(+1)$*linha de a_1	0	-1	0	3	1	1	1	0	12
$(+1)$*linha de a_2	0	0	-1	2	2	0	0	1	10
z_a	1	-1	-1	5	3	1	0	0	22

- O quadro seguinte é válido:

Voltar

	z_a	y_1	y_2	y_3	y_4	y_5	a_1	a_2	
a_1	0	-1	0	3	1	1	1	0	12
a_2	0	0	-1	2	2	0	0	1	10
z_a	1	-1	-1	5	3	1	0	0	22

	z	y_1	y_2	y_3	y_4	y_5	
y_3	0	-1/2	1/4	1	0	1/2	7/2
y_4	0	1/2	-3/4	0	1	-1/2	3/2
z	1	0	0	-120	-80	-30	0

- Exprimir a função objectivo z em função das variáveis não-básicas y_1, y_2 e y_5 usando eliminação de Gauss: somar à linha de z as linhas de y_3 e y_4 multiplicadas por constantes adequadas.

	z	y_1	y_2	y_3	y_4	y_5	
(+1)*linha de z	1	0	0	-120	-80	-30	0
(+120)*linha de y_3	0	-60	30	120	0	60	420
(+80)*linha de y_4	0	40	-60	0	80	-40	120
z	1	-20	-30	0	0	-10	540

- O quadro seguinte é válido:

Voltar

	z	y_1	y_2	y_3	y_4	y_5	
y_3	0	-1/2	1/4	1	0	1/2	7/2
y_4	0	1/2	-3/4	0	1	-1/2	3/2
z	1	-20	-30	0	0	-10	540

Fim