Métodos de Apoio à Decisão

Propriedades da Dualidade em Otimização Linear

João Pedro Pedroso

2024/2025

Aula passada: Noções estudadas

- Problemas de maximização normal e de minimização normal
- Dual de um problema linear
- Interpretação económica da dualidade.
- Noções de preço sombra e de custo reduzido.

Aula passada: primal/dual; para um problema linear normal:

Primal (n variáveis, m restrições):

$$
\begin{array}{rcl}
\max z = & c_1 x_1 + c_2 x_2 + \ldots + c_n x_n & \\
\text{sujeito a} & a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n & \leq b_1 \\
& a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n & \leq b_2 \\
& \ldots & \\
& a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n & \leq b_m \\
& x_1, x_2, \ldots, x_n & \geq 0
\end{array}
$$

Dual (m variáveis, n restrições):

$$
\begin{array}{rcl}
\min w = & b_1 y_1 + b_2 y_2 + \ldots + b_m y_m & \\
\text{sujeito a} & a_{11} y_1 + a_{21} y_2 + \ldots + a_{m1} y_m & \geq c_1 \\
& a_{12} y_1 + a_{22} y_2 + \ldots + a_{m2} y_m & \geq c_2 \\
& \ldots & \\
& a_{1n} y_1 + a_{2n} y_2 + \ldots + a_{mn} y_m & \geq c_n \\
& y_1, y_2, \ldots, y_m & \geq 0
\end{array}
$$

Exemplo: solução ótima:

Primal:

$$
\begin{array}{rlll}
\text{maximizar } z = & 60x_1 + 30x_2 + 20x_3 & & \\
\text{sujeito a} & 8x_1 + 6x_2 + x_3 & \leq 48 & (\textit{tabuas}) \\
& 4x_1 + 2x_2 + 1.5x_3 & \leq 20 & (\textit{acabamentos}) \\
& 2x_1 + 1.5x_2 + 0.5x_3 & \leq 8 & (\textit{carpintaria}) \\
& x_1, x_2, x_3 & \geq 0 &
\end{array}
$$

$z = 280, x_1 = 2, x_2 = 0, x_3 = 8, s_1 = 24, s_2 = 0, s_3 = 0$

Dual:

$$
\begin{array}{rlll}
\text{minimizar } w = & 48y_1 + 20y_2 + 8y_3 & & \\
\text{sujeito a} & 8y_1 + 4y_2 + 2y_3 & \geq 60 & (\textit{secretarias}) \\
& 6y_1 + 2y_2 + 1.5y_3 & \geq 30 & (\textit{mesas}) \\
& y_1 + 1.5y_2 + 0.5y_3 & \geq 20 & (\textit{cadeiras}) \\
& y_1, y_2, y_3 & \geq 0 &
\end{array}
$$

$w = 280, y_1 = 0, y_2 = 10, y_3 = 10, e_1 = 0, e_2 = 5, e_3 = 0$

Aula de hoje:

- propriedades da dualidade
- gestão de projetos

Propriedades da dualidade

Considere-se o **primal** como um problema normal de maximização:

$$\begin{array}{rlll} \max z = & \sum_{j=1}^{n} c_j x_j & & \\ \text{sujeito a} & \sum_{j=1}^{n} a_{ij} x_j & \leq b_i & i = 1, 2, \ldots, m \\ & x_j & \geq 0 & j = 1, 2, \ldots, n \end{array}$$

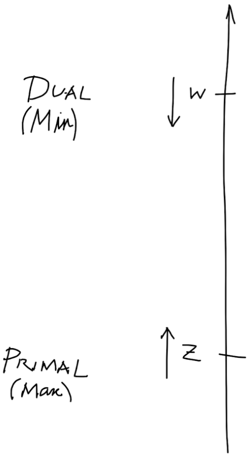
O problema **dual** é o seguinte:

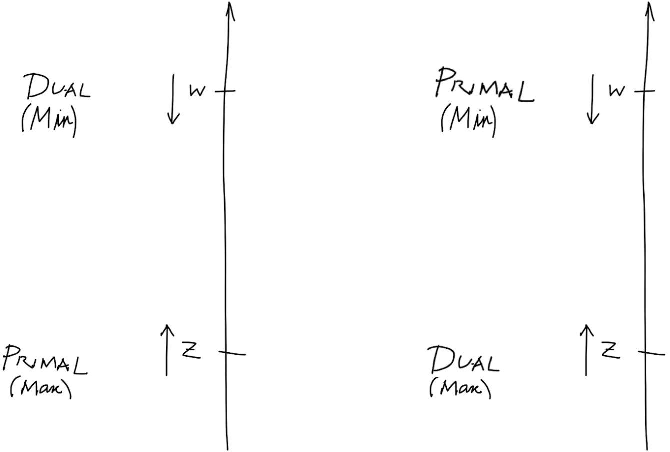
$$\begin{array}{rlll} \min w = & \sum_{i=1}^{m} b_i y_i & & \\ \text{sujeito a} & \sum_{i=1}^{m} a_{ij} y_i & \geq c_j & j = 1, 2, \ldots, n \\ & y_i & \geq 0 & i = 1, 2, \ldots, m \end{array}$$

Propriedade 1

Dualidade fraca
Se x (vetor de dimensão n) é uma solução admissível do primal e y (vetor de dimensão m) é uma solução admissível do dual, então

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} y_i b_i$$

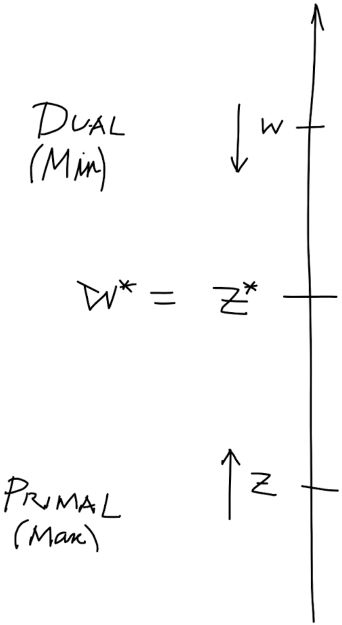

DUAL
(Min)
w
PRIMAL
(Max)
z


DUAL
(Min)
w
PRIMAL
(Max)
z
PRIMAL
(Min)
w
DUAL
(Max)
z

Propriedade 2

Dualidade forte
Se x^* é uma solução ótima do primal e y^* é uma solução ótima do dual, então

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} y_i^* b_i$$


DUAL
(Min)
w
w* = z*
PRIMAL
(Max)
z

Propriedade 3

Se o primal é ilimitado, então o dual é impossível

Suponhamos que o dual não é impossível; então:

$$\exists y^\circ : w^\circ = \sum_{i=1}^{m} y_i^\circ b_i$$

com y° admissível. Então, pela dualidade fraca $\sum_{j=1}^{n} c_j x_j \leq w^\circ$, ou seja, o primal não é ilimitado

Propriedade 4

Se o dual é ilimitado, então o primal é impossível
(Pode acontecer que tanto o primal como o dual sejam impossíveis.)

Propriedade 5

Dupla otimalidade para qualquer par de problemas duais:

- a existência de solução ótima (finita) para um deles garante a existência de solução ótima (finita) o outro
- nesse caso, o ótimo é o mesmo: $z^* = w^*$.

Propriedade 6

Simetria para qualquer par de problemas lineares primal e dual, todas as relações entre eles são simétricas, porque o dual deste problema dual é este problema primal.

Teorema da dualidade: possibilidades de relação entre primal e dual:

		Primal	
		Possível	Impossível
DUAL	Possível	$z^* = w^*$	$w^* \to \infty$
	Impossível	$z^* \to \infty$	ambos impossíveis

Propriedade dos desvios complementares

Seja o par de problema duais possíveis:

$$
\begin{array}{llll}
\max z = & \sum_{j=1}^{n} c_j x_j & & \\
\text{sujeito a} & \sum_{j=1}^{n} a_{ij} x_j & \leq b_i & i = 1, 2, \dots, m \\
& x_j & \geq 0 & j = 1, 2, \dots, n
\end{array}
\qquad
\begin{array}{llll}
\min w = & \sum_{i=1}^{m} b_i y_i & & \\
& \sum_{i=1}^{m} a_{ij} y_i & \geq c_j & j = 1, 2, \dots, n \\
& y_i & \geq 0 & i = 1, 2, \dots, m
\end{array}
$$

- Seja s_i $(i = 1, \dots, m)$ a variável de desvio associada à restrição i (primal: quantidade de recurso i não utilizada)
- Seja e_j $(j = 1, \dots, n)$ a variável de desvio associada à restrição j (dual: custo reduzido associado à produção de j)
- Na solução ótima (x^*, y^*, s^*, e^*) verifica-se que:
  - $s_i^* y_i^* = 0$, para $i = 1, \dots, m$
  - $e_j^* x_j^* = 0$, para $j = 1, \dots, n$
- Ou seja, na solução ótima:
  - $s_i^* > 0 \Rightarrow y_i^* = 0$ e $y_i^* > 0 \Rightarrow s_i^* = 0$
  - $e_j^* > 0 \Rightarrow x_j^* = 0$ e $x_j^* > 0 \Rightarrow e_j^* = 0$

Propriedade 8

Se x e y são

- soluções admissíveis para o primal e dual, respetivamente
- verificam a propriedade dos desvios complementares

então x e y são as **soluções ótimas** do primal e dual, respetivamente.

Propriedade da simetria e teorema da dualidade

Propriedade da simetria: para qualquer par de problemas lineares primal e dual, as relações entre eles são simétricas: o dual deste problema dual é este problema primal.

Teorema da dualidade: relações possíveis entre o primal e o dual:

1. Se um dos problemas tem soluções admissíveis e um objetivo limitado (i.e., tem solução ótima), então o outro problema também. Neste caso, a dualidade forte e a dualidade fraca aplicam-se.
2. Se um dos problemas tem soluções admissíveis e um objetivo ilimitado (não solução ótima), então o outro problema não tem soluções admissíveis (é impossível).
3. Se um problema não tem soluções admissíveis (é impossível), então o outro ou é impossível ou tem o objetivo ilimitado.

Exemplo

Problema de dietas. Considere o problema da escolha de alimentos, de forma a satisfazer certas exigências nutricionais. Os pratos disponíveis e os respetivos preços estão indicados na tabela abaixo. Cada prato fornece as seguintes percentagens (por prato) dos mínimos diários necessários em calorias, proteínas, açucares, e lípidos:

	Preço	calorias	proteínas	açúcares	lípidos
Sopa	200	20	20	10	10
Bife	500	40	30	10	15
Gelado	300	40	10	25	25

O problema é encontrar a combinação de pratos que satisfaça as exigências alimentares de um dia (100% do mínimo diário) com o custo mínimo.

1. Formule o problema.
2. Utilizando o GLPK, resolva o problema.
3. Determine os valores no ótimo de variáveis, variáveis duais, custos reduzidos, e variáveis desvio.
4. Formule o problema dual.
5. Resolva os problema dual e determine os valores no ótimo de variáveis, variáveis duais, custos reduzidos, e variáveis desvio.
6. Verifique que o teorema dos desvios complementares se aplica, e dê-lhe uma interpretação económica.

Exemplo: modelo

Problema de dietas

```
set PRATOS;      # conjunto de pratos disponiveis
set NUTR;        # nutrientes que interessa garantir

param preco{PRATOS};     # preco de cada prato
param composicao{NUTR, PRATOS}; # conteudo em cada
                # nutriente existente em cada prato
                # (em % do minimo diario)
param min_nutriente{NUTR};       # quantidade necessaria
                # de cada nutriente

var Quant{PRATOS}>=0;    # numero de vezes que se come
                         # cada prato

subject to Requisito{i in NUTR}:
        sum {j in PRATOS} composicao[i,j]*Quant[j]
        >= min_nutriente[i];

minimize custo: sum {j in PRATOS} preco[j]*Quant[j];
```

Exemplo: dados

```
set NUTR := CALORIAS PROTEINAS LIPIDOS ACUCARES;
set PRATOS := ARROZ, SANDWICH, BOLO;

param: preco :=
        ARROZ    500
        SANDWICH 250
        BOLO     150;

param: min_nutriente :=
   CALORIAS     750
   PROTEINAS      6
   LIPIDOS       10
   ACUCARES       8    ;

param composicao:
              ARROZ     SANDWICH     BOLO :=
   CALORIAS     450       250        170
   PROTEINAS      3         2          0
   LIPIDOS        2         4          1
   ACUCARES       2         2          4;
```

```
Problem:    teorica
Rows:       5
Columns:    3
Non-zeros:  14
Status:     OPTIMAL
Objective:  custo = 825 (MINimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 Requisito[CALORIAS]
                    B            835           750
     2 Requisito[PROTEINAS]
                    NL             6             6                        87.5
     3 Requisito[LIPIDOS]
                    B           12.5            10
     4 Requisito[ACUCARES]
                    NL             8             8                        37.5
     5 custo        B            825

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 Quant[ARROZ] NL             0             0                       162.5
     2 Quant[SANDWICH]
                    B              3             0
     3 Quant[BOLO]  B            0.5             0

Karush-Kuhn-Tucker optimality conditions:
```

Problemas de gestão de projetos

- Modelos mais utilizados:
  - CPM: *critical path method* $\rightarrow$ quando a duração de cada atividade é conhecida com certeza (de forma determinística). Determina a duração mínima do projeto e as atividades críticas.
  - PERT: *program evaluation and review technique* $\rightarrow$ quando a duração das atividades tem uma componente estocástica. Permite estimar a probabilidade de o projeto ser concluído num determinado prazo.

Exemplo

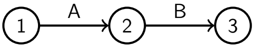
Uma companhia de software pretende introduzir no mercado um novo programa (programa 3), que consiste na integração de uma base de dados contabilísticos (programa 1) com um programa de análise financeira (programa 2), que deverão ser reunidos num ambiente gráfico coerente. Antes de começar a programação dos programas 1 e 2, terá de ser feita a escolha e compra dos compiladores da linguagem que se quer utilizar, e de se fazer um treino dos programadores para relembrarem as especificidades dessa linguagem. Antes de se poder integrar os programas 1 e 2, deverá ser feito um teste exaustivo e debugging do programa 2. A lista de atividades e as precedências são as seguintes:

Atividade	Descrição	Precedências	Duração
A	Treino de programadores	—	6
B	Compra de compiladores	—	9
C	Programa 1	A,B	18
D	Programa 2	A,B	7
E	Testar programa 2	D	10
F	Integrar programas 1 e 2	C,E	12

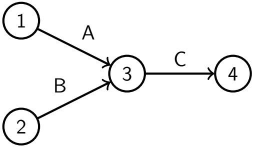
CPM: critical path method

- Requisitos:
  - a lista de atividades de constituem o projeto
  - para cada atividade: conjunto de atividades que a deverá preceder (relações de precedência)
- Conclusão do projeto: quanto todas as atividades terminarem
- Modelo de rede com atividades nos arcos:
  - *atividades* representadas por *arcos* num grafo dirigido
  - *vértices:* representam *acontecimentos* (instantes no tempo)

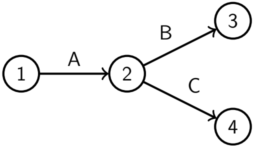
- A atividade A precede B (i.e., B depende de A):

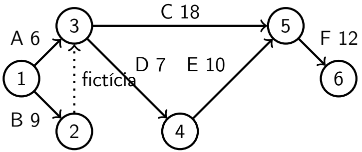


- C depende de A e de B:



- B e C dependem de A:





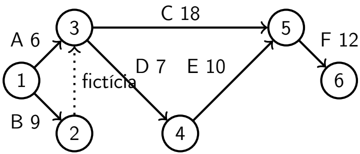
1. Nó 1 representa o começo do projeto; todas as atividades que não têm precedentes partem do nó 1.
2. Um nó que representa a conclusão do projeto (nó terminal) deverá ser incluído.
3. Numerar os nós por forma a que o nó que representa a conclusão de uma atividade tenha sempre um número maior que o do seu início.
4. Cada atividade deverá ter exatamente um arco a representá-la.
5. Dois nós podem estar ligados por um arco, no máximo (para isso pode ser necessário criar atividades fictícias).

Nota: Ao desenhar a rede de atividades, devemos utilizar apenas os *precedentes diretos* de cada atividade.

Datas mais cedo e mais tarde

- **Data mais cedo** *(earliest start)* para o acontecimento i, DMC(i): período de tempo mínimo entre o início do projeto e a conclusão de todas as atividades que convergem em i.
  1. $\mathrm{DMC}(1) = 0$
  2. determinar todos os acontecimentos que precedem i
  3. para cada um deles, calcular $\mathrm{DMC}(j) + t_{ji}$
  4. $\mathrm{DMC}(i) =$ máximo das somas calculadas no passo anterior
- **Data mais tarde** *(latest start)* para o acontecimento i, DMT(i): momento mais tardio de ocorrência de i, por forma a que não haja atrasos na conclusão do projeto.
  1. DMT(nó terminal) = DMC(nó terminal)
  2. determinar todos os acontecimentos que sucedem a i
  3. para cada um deles, calcular $\mathrm{DMT}(j) - t_{ij}$
  4. $\mathrm{DMT}(i) =$ mínimo das diferenças calculadas no passo anterior

Exemplo



1. $\mathrm{DMC}(1) = 0$
2. $\mathrm{DMC}(2) = 0 + 9 = 9$
3. $\mathrm{DMC}(3) = \max(0+6, 9+0) = 9$
4. $\mathrm{DMC}(4) = 9 + 7 = 16$
5. $\mathrm{DMC}(5) = \max(9 + 18, 16 + 10) = 27$
6. $\mathrm{DMC}(6) = 27 + 12 = 39$

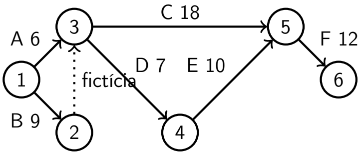
1. $\mathrm{DMT}(6) = 39$
2. $\mathrm{DMT}(5) = 39 - 12 = 27$
3. $\mathrm{DMT}(4) = 27 - 10 = 17$
4. $\mathrm{DMT}(3) = \min(27 - 18, 17 - 7) = 9$
5. $\mathrm{DMT}(2) = 9 - 0 = 9$
6. $\mathrm{DMT}(1) = \min(9 - 9, 9 - 6) = 0$

Folga total

- Folga total para o arco (i,j): valor máximo que a atividade (i,j) se pode atrasar por forma a que a duração total do projeto não seja alterada (mantendo todas as outras atividades a tempo).
- Seja t_{ij} a duração da atividade (i,j):
  - a atividade começa em i
  - se se atrasar o início de (i,j) em k unidades, o projeto não se atrasa sse $\mathrm{DMC}(i) + k + t_{ij} \leq \mathrm{DMT}(j)$
  - k máximo é a folga total da atividade (i,j):

$$\mathrm{FT}(i,j) = \mathrm{DMT}(j) - \mathrm{DMC}(i) - t_{ij}$$

No exemplo anterior

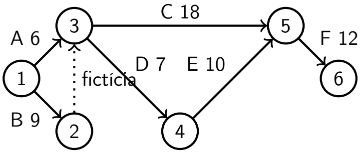


Atividade A	$\text{FT}(1,3) = \text{DMT}(3) - \text{DMC}(1) - 6 = 3$
Atividade B	$\text{FT}(1,2) = \text{DMT}(2) - \text{DMC}(1) - 9 = 0$
Atividade C	$\text{FT}(3,5) = \text{DMT}(5) - \text{DMC}(3) - 18 = 0$
Atividade D	$\text{FT}(3,4) = \text{DMT}(4) - \text{DMC}(3) - 7 = 1$
Atividade E	$\text{FT}(4,5) = \text{DMT}(5) - \text{DMC}(4) - 10 = 1$
Atividade F	$\text{FT}(5,6) = \text{DMT}(6) - \text{DMC}(5) - 12 = 0$
Atividade fictícia	$\text{FT}(2,3) = \text{DMT}(3) - \text{DMC}(2) - 0 = 0$

Atividades e caminhos críticos

- **Atividade crítica:** atividade com folga $FT = 0$.
- **Caminho crítico:** caminho entre o nó 1 e o nó terminal que consiste *apenas* em atividades críticas.
- Para uma atividade crítica, se se aumentar a sua duração em Δ unidades, o projeto será atrasado em Δ unidades.

Formulação em programação linear



$$
\begin{array}{lrll}
\text{minimizar} & & x_6 - x_1 & \\
\text{sujeito a} & : & x_3 \geq x_1 + 6 & (A) \\
& & x_2 \geq x_1 + 9 & (B) \\
& & x_3 \geq x_2 & (fict) \\
& & x_5 \geq x_3 + 18 & (C) \\
& & x_4 \geq x_3 + 7 & (D) \\
& & x_5 \geq x_4 + 10 & (E) \\
& & x_6 \geq x_5 + 12 & (F) \\
& & x_i \geq 0, \forall i &
\end{array}
$$

Análise

- Atividades críticas: variável dual correspondente é diferente de zero.
- Caminho crítico: caminho entre o começo e a conclusão do projeto, correspondente às atividades críticas.
- Exemplo anterior: ver `cpm.mod`, `cpm.sol`.

cpm.mod

```
var x1 >=0;
var x2 >=0;
var x3 >=0;
var x4 >=0;
var x5 >=0;
var x6 >=0;

minimize z: x6 - x1;

A: x3 >= x1 + 6;
B: x2 >= x1 + 9;
fict1: x3 >= x2;
C: x5 >= x3 + 18;
D: x4 >= x3 + 7;
E: x5 >= x4 + 10;
F: x6 >= x5 + 12;
```

cpm.sol

```
Problem:    cpm
Rows:       8
Columns:    6
Non-zeros:  16
Status:     OPTIMAL
Objective:  z = 39 (MINimum)
   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 z            B             39
     2 A            B              9             6
     3 B            NL             9             9                           1
     4 fict1        NL             0            -0                           1
     5 C            NL            18            18                           1
     6 D            NL             7             7                       < eps
     7 E            B             11            10
     8 F            NL            12            12                           1
   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 x1           NL             0             0                       < eps
     2 x2           B              9             0
     3 x3           B              9             0
     4 x4           B             16             0
     5 x5           B             27             0
     6 x6           B             39             0
```

Noções estudadas

- Propriedades da dualidade (conclusão).
- Preço sombra.
- Custo reduzido.
- Problemas de gestão de projetos:
  - *CPM*
  - caminho crítico, atividade crítica
  - formulação em programação matemática

Próxima aula

- Problemas de gestão de projetos:
  - *PERT*
  - simulação.