

# Programação Linear - Introdução

## Investigação Operacional

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

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

5 de setembro de 2019

- Definições de Investigação Operacional
- Metodologia da Investigação Operacional
- Modelos de Programação Linear
- Exemplos
- Programação Linear: algumas áreas de aplicação
- Programação Linear: alguns Marcos

# Definições de Investigação Operacional

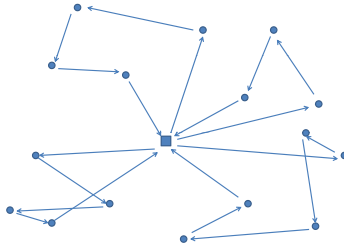
- A Investigação Operacional no sentido mais lato pode ser caracterizada como sendo a aplicação de métodos, técnicas e ferramentas científicas a problemas que envolvem a operação de sistemas, por forma a prover os responsáveis pelo controlo da operação com soluções óptimas para os problemas (*Churchman, Ackoff, Arnoff, 1957*).
- A Investigação Operacional pode ser descrita como uma abordagem científica da tomada de decisões que envolvem a operação de um sistema organizacional (*Hillier, Lieberman, 1974*).
- A Investigação Operacional é uma abordagem científica de resolução de problemas de gestão (*Wagner, 1975*).

- Identificar o problema a abordar.
- Observar o sistema e recolher dados.
- Formular um modelo matemático do problema.
- Verificar e validar o modelo.
- Seleccionar uma decisão adequada.
- Implementar e avaliar.

# Exemplo: problema de planeamento de rotas de veículos

## Objectivo

- Dados um conjunto de veículos com capacidades e
- um conjunto de clientes com procuras e janelas temporais de visita, encontrar a solução de custo mínimo, em que haja
- um conjunto de rotas, todas começando e terminando no depósito,
- sendo cada cliente visitado por um único veículo.



# Construção de modelos de Investigação Operacional (IO)

Após identificar o problema, a observação do sistema permite reconhecer:

- a forma de representar uma decisão admissível,
- as regras gerais que determinam quais as decisões admissíveis, e
- a forma de atribuir um valor a cada decisão.

## Modelos de Investigação Operacional

- Um modelo matemático identifica as decisões admissíveis.
- Para associar um valor a cada decisão, há uma *medida de eficiência*,
- que permite identificar a(s) melhor(es) decisão(ões).

Os modelos de *Programação Linear* são os mais usados em IO:

- *programação* é usado no sentido de planeamento, como em programação de actividades, e
- as funções envolvidas nos modelos são *lineares*.

# Programação Linear: elementos dos modelos

## Variáveis de decisão:

- são incógnitas que associamos às decisões (soluções) admissíveis.
- e.g.,  $x_1$  - quantidade de produto 1 a fabricar numa semana.

## Parâmetros:

- são dados do sistema que não podem ser alterados.
- e.g., volume de mão-de-obra disponível por semana.

# Programação Linear: estrutura dos modelos

## Restrições:

- definem o conjunto de decisões (soluções) admissíveis.
- Uma restrição exprime uma relação entre uma **função linear** das variáveis de decisão e uma constante (e.g., um parâmetro).
- e.g.,  $3x_1 + 2x_2 \leq 120$ .

## Função objectivo:

- (ou medida de eficiência) é uma **função linear** das variáveis de decisão que associa um valor a cada solução do conjunto de soluções admissíveis.
- e.g., a função linear  $12x_1 + 10x_2$  associa um valor a cada solução.

## Fim em vista:

- determinar a(s) solução(ões) que optimiza(m) a função objectivo.
  - Maximizar lucro
  - Minimizar custo



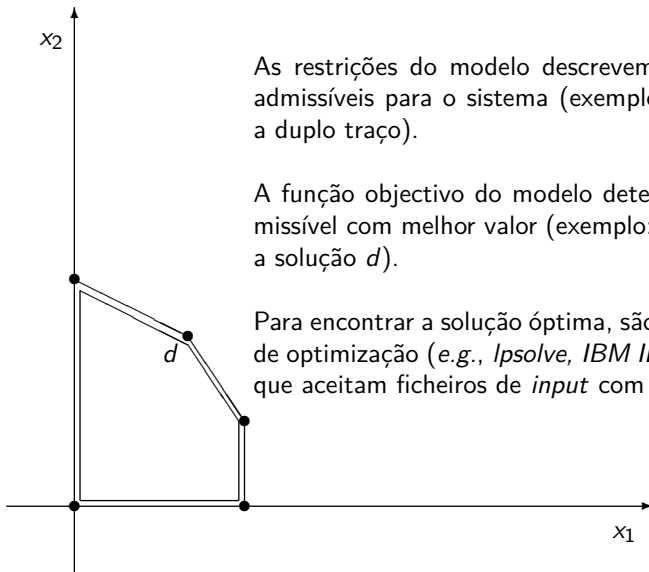
# Programação linear: forma geral do modelo

- Problema de optimização (ou maximização ou minimização) de uma função objectivo linear sujeita a restrições lineares:

$$\begin{aligned} \{\max, \min\} z &= \sum_{j=1}^n c_j x_j \\ \text{sujeito a} \quad &\sum_{j=1}^n a_{ij} x_j \leq b_i, \forall i \in R_{\leq} \\ &\sum_{j=1}^n a_{ij} x_j \geq b_i, \forall i \in R_{\geq} \\ &\sum_{j=1}^n a_{ij} x_j = b_i, \forall i \in R_{=} \\ &x_j \geq 0, \forall j \end{aligned}$$

- $z$ : função objectivo
- $x_1, \dots, x_j, \dots, x_n$ : variáveis de decisão.
- $c_1, \dots, c_j, \dots, c_n$ : coeficientes da função objectivo  $\sum_{j=1}^n c_j x_j$ .
- $b_i$ : coeficiente do lado direito das restrições, indexado por  $i$
- $x_j \geq 0$ : restrições de não-negatividade, para todos os  $j$ .

# Programação Linear: modelo e resolução do modelo



As restrições do modelo descrevem **todas** as soluções admissíveis para o sistema (exemplo: região delimitada a duplo traço).

A função objectivo do modelo determina a solução admissível com melhor valor (exemplo: a solução óptima é a solução  $d$ ).

Para encontrar a solução óptima, são usadas ferramentas de optimização (e.g., *Ipsolve*, *IBM ILOG Cplex*, *Gurobi*), que aceitam ficheiros de *input* com formatos definidos.

[▶ Ver mais](#)

# Programação Linear: elementos dos modelos (exemplos)

- Variáveis de decisão:
  - quantidades a produzir de um artigo
  - rotas a percorrer por veículos
  - fluxos a enviar pelos arcos duma rede
  - actividades (ou projectos) a seleccionar
  - instantes para iniciar as actividades
- Parâmetros
  - recursos disponíveis
  - número de unidades pedidas por um cliente
  - tempo de processamento de uma actividade
- Restrições:
  - relação entre a função que exprime a quantidade de um recurso que as actividades consomem e o recurso disponível
  - relação entre a função que exprime a quantidade de um bem que as actividade produzem e a procura a satisfazer
  - relação que traduz as regras de funcionamento do sistema (e.g., conservação de fluxo, precedência entre operações)

# Exemplo 1: problema de produção

- Uma empresa produz 2 tipos de artigos: Artigo 1 e Artigo 2.
- A produção destes artigos requer 3 tipos de recursos:
  - Material
  - Mão de Obra
  - Tempo-Máquina
- Objectivo: determinar o **plano de produção diário** (solução admissível) que **maximiza o lucro total** (com o valor óptimo).
- A quantidade disponível de cada recurso, o consumo de recursos por cada artigo produzido e o lucro líquido de cada artigo são:

	Artigo 1	Artigo 2	Quantidade disponível
Material	3 [unid./art.]	2 [unid./art.]	120 [unid./dia]
Tempo-Homem	1 [h.hom./art.]	2 [h.hom./art.]	80 [h.hom./dia]
Tempo-Máquina	1 [h.maq./art.]	0 [h.maq./art.]	30 [h.maq./dia]
Lucro Unitário	12 [U.M./art.]	10 [U.M./art.]	

# Exemplo 1: problema de produção - elementos do modelos

## Variáveis de decisão (incógnitas associadas às decisões admissíveis):

- $x_1$  : quantidade de artigos de tipo 1 a fabricar diariamente [art./dia]
- $x_2$  : quantidade de artigos de tipo 2 a fabricar diariamente [art./dia]

## Parâmetros (dados do sistema que não podem ser alterados):

- quantidade disponível de cada recurso;
- lucro unitário dos artigos;
- consumo de recursos por cada artigo (coeficientes tecnológicos)

# Exemplo 1: problema de produção - restrições e função objectivo

## Restrições:

- função linear  $3x_1 + 2x_2$  : qtd. de material usada diariamente [unid./dia]
- restrição  $3x_1 + 2x_2 \leq 120$  : apenas são admissíveis as soluções que não excedam a disponibilidade diária de material [unid./dia]
- as outras restrições são semelhantes.
- Todas as variáveis têm restrições de não-negatividade  $(x_1, x_2 \geq 0)$ .(\*)

## Função objectivo:

- função objectivo  $12x_1 + 10x_2$  : lucro diário [U.M./dia]

(\*) o caso em que as variáveis podem ser positivas ou negativas (s/ restrição de sinal) está tratado nos diapositivos

Transformações Básicas.

# Exemplo 1: problema de produção - modelo

- Função objectivo:

$$\max z = 12x_1 + 10x_2$$

- Restrições:

$$3x_1 + 2x_2 \leq 120$$

$$1x_1 + 2x_2 \leq 80$$

$$1x_1 \leq 30$$

$$x_1, x_2 \geq 0$$

# Exemplo 1: problema de produção - resolução

- Resolvendo o seguinte modelo com um *solver* de PL:

```
max: 12x1 + 10x2;  
material: 3x1 + 2x2 <= 120;  
maodobra: 1x1 + 2x2 <= 80;  
tmaquina: 1x1          <= 30;
```

- obtém-se o seguinte relatório com a solução óptima:

Objective	
Variables	result
	540
x1	20
x2	30

- que é fazer 20 unid./dia de produto 1 e 30 unid./dia de produto 2, com um lucro diário de 540 U.M..



## Exemplo 2: sudoku $2 \times 2$

- O sudoku é um puzzle lógico em que se pretende preencher todas as células com algarismos, de forma a que não haja repetição de nenhum algarismo nas linhas, nas colunas ou nos blocos.
- Vamos ver uma versão com uma matriz  $4 \times 4$ , dividida em 4 blocos  $2 \times 2$ , em que os algarismos de algumas das células são dados.

Exemplo:

	1	4	
			2
1			
	3	2	

- Como representar uma decisão? Quais as variáveis de decisão?
- Quais as restrições?
- Qual é a função objectivo?
- Qual o espaço de soluções admissíveis?

## Exemplo 2: sudoku 2 × 2 - elementos do modelo

Variáveis de decisão:

$$x_{ijk} = \begin{cases} 1 & , \text{ se a célula } (i,j) \text{ tiver o algarismo } k \\ 0 & , \text{ caso contrário} \end{cases}$$

Restrições:

- cada célula apenas pode ter um algarismo;
- numa linha, cada algarismo apenas pode aparecer uma vez;
- numa coluna, cada algarismo apenas pode aparecer uma vez;
- num bloco 2 × 2, cada algarismo apenas pode aparecer uma vez; e
- há células que devem ter o algarismo dado no puzzle.

Função objectivo:

- O objectivo do jogo é obter uma solução admissível,
- sendo todas as soluções igualmente boas.
- A função objectivo pode ser uma função qualquer.

## Exemplo 2: sudoku 2 × 2 - solução do modelo

### Modelo:

- ver ficheiro sudoku\_2x2.lp

### Solução do modelo, obtida com o *lpsolve*:

- $x_{121} = x_{134} = x_{242} = x_{311} = x_{423} = x_{432} = 1$  (conforme atribuição feita no modelo)
- $x_{112} = x_{143} = x_{213} = x_{224} = x_{231} = x_{322} = x_{333} = x_{344} = x_{414} = x_{441} = 1$
- todas as outras variáveis  $x_{ijk} = 0$

### Solução do puzzle:

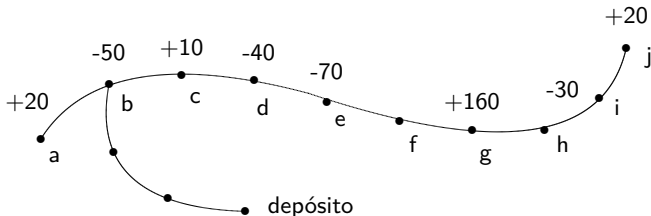
2	1	4	3
3	4	1	2
1	2	3	4
4	3	2	1

## Exemplo 3: transporte de terras

- As obras de terraplanagem representam uma parte significativa dos custos de construção de vias de comunicação.
- Grandes volumes de terra devem ser deslocados de zonas de empréstimo para zonas de depósito para obter os nivelamentos desejados.
- Os custos de transporte de terra são aproximadamente proporcionais à distância percorrida.
- Objectivo: estabelecer o plano de deslocação de terra para minimizar custos de terraplanagem.

## Exemplo 3: transporte de terras (cont.)

- Os pontos  $a, b, \dots, j$  encontram-se distanciados entre si de 10 Km.
- As quantidades associadas aos pontos indicam os volumes de terra a deslocar (em milhares de  $m^3$ ), sendo as zonas de empréstimo e de depósito assinaladas pelos sinais + e -, respectivamente.
- Pode ainda recorrer-se a uma zona de depósito, situada fora do traçado da via, a uma distância de 30 Km do ponto  $b$ .



- Quais as variáveis de decisão? Quais os dados? Quais as restrições?

# Exemplo 3: transporte de terras - elementos dos modelos

## Variáveis de decisão:

- volume de terra a transportar desde cada zona de empréstimo para cada zona de depósito.

## Parâmetros:

- volume de terra a deslocar de cada zona;
- distâncias;
- custo de transporte por  $m^3$  e por Km.

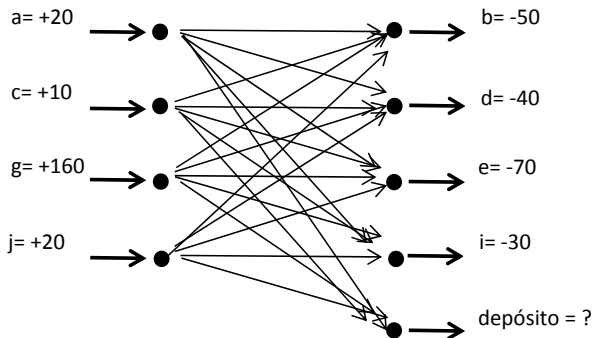
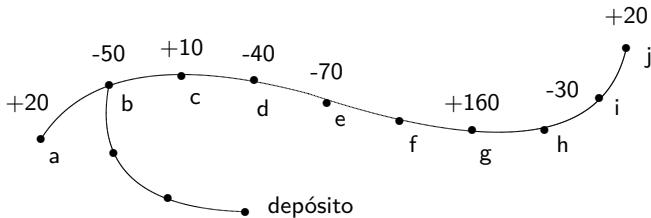
## Restrições:

- a soma dos volumes de terra que saem de uma dada zona de empréstimo para cada zona de depósito perfaz o volume a deslocar;
- a soma dos volumes de terra que chegam a uma dada zona de depósito de cada zona de empréstimo perfaz o volume a depositar;

## Função objectivo:

- função de custo de transporte.

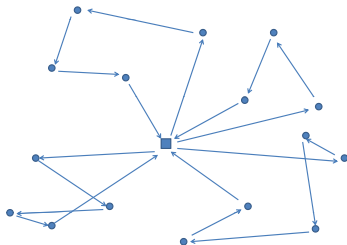
# Exemplo 3: transporte de terras - esboço de modelo



# Exemplo 4: planeamento de rotas de veículos

## Objectivo

- Dados um conjunto de veículos com capacidades e
- um conjunto de clientes com procuras e janelas temporais de visita, encontrar a solução de custo mínimo, em que haja
- um conjunto de rotas, todas começando e terminando no depósito,
- sendo cada cliente visitado por um único veículo.



- Quais as variáveis de decisão? Quais os dados? Quais as restrições?



# Exemplo 4: planeamento de rotas de veículos - elementos

Há modelos diferentes consoante a escolha das variáveis de decisão.

Variáveis de decisão (um exemplo):

- cada rota (uma sequência de arcos) admissível.

Parâmetros:

- carga para cada cliente; capacidade dos veículos;
- distâncias e tempos de viagem;
- ...

Restrições:

- carga não excede a capacidade do veículo;
- cada cliente é visitado por um único veículo na sua janela temporal;
- ...

Função objectivo:

- função de custo de transporte.

# Programação linear: algumas áreas de aplicação

- Logística e distribuição
- Telecomunicações e redes de comunicação
- Gestão da cadeia de abastecimento
- Gestão de serviços de saúde
- Planeamento da operação de companhias de transporte (aéreo, caminho de ferro, urbano)
- Planeamento da produção
- Gestão de projectos
- Corte e empacotamento
- Gestão de pessoal
- Gestão de florestas
- ...

# Programação linear: alguns marcos

- Gauss: Eliminação de Gauss para resolver um sistema de equações
  - Kantorovich (1930) : atribuição eficiente de recursos, e.g., modelos de planeamento industrial e modelos de corte (galardoado com o prémio Nobel (1975) juntamente com Tim Koopmans)
  - Dantzig (1947) : método Simplex
  - Khachian (1979) : método do elipsóide
  - Karmarkar (1984) : método de pontos interiores
- 
- Simplex algorithm: one of the top ten algorithms of the 20th century (The Best of the 20th Century: Editors Name Top 10 Algorithms, Barry A. Cipra, from SIAM News, Volume 33, Number 4)

- A programação linear é uma ferramenta de uso generalizado na indústria e nos serviços.
- A sua utilização traduz-se em economias muito relevantes.

# Resultados de aprendizagem

- Definir a Investigação Operacional
- Descrever a metodologia da investigação operacional
- Identificar a estrutura dos modelos de programação linear
- Desenvolver a capacidade de analisar sistemas complexos e de criar modelos para os descrever.
  - identificar alternativas de decisão e objectivos em problemas de decisão;
  - identificar variáveis de decisão;
  - identificar parâmetros (dados)



# (Alguns) solvers de programação matemática

---

CPLEX	<a href="https://www.ibm.com/analytics/cplex-optimizer">https://www.ibm.com/analytics/cplex-optimizer</a>
GUROBI	<a href="https://www.gurobi.com/">https://www.gurobi.com/</a>
XPRESS-MP	<a href="http://www.dashoptimization.com">http://www.dashoptimization.com</a>
COIN CLP	<a href="http://www.coin-or.org">http://www.coin-or.org</a>
GLPK	<a href="http://www.gnu.org/software/glpk">http://www.gnu.org/software/glpk</a>
LP SOLVE	<a href="https://sourceforge.net/projects/lpsolve/">https://sourceforge.net/projects/lpsolve/</a>

---

Ver também:

- Software Survey: Linear Programming, Robert Fourer

<https://www.informs.org/ORMS-Today/Public-Articles/June-Volume-44-Number-3/Linear-Programming-Software-Survey>

# (Algumas) linguagens de modelação

AMPL	A Modeling Language for Mathematical Programming	<a href="http://www.ampl.com">www.ampl.com</a>	*
GAMS	General Algebraic Modeling System	<a href="http://www.gams.com">www.gams.com</a>	*
OPL	Optimization Programming Language	<a href="http://www.ilog.com">www.ilog.com</a>	*
LINGO	Lingo	<a href="http://www.lindo.com">www.lindo.com</a>	*
GNU-MP	GNU Mathematical Programming Language	<a href="http://www.gnu.org/software/glpk">www.gnu.org/software/glpk</a>	+

(\*) - comercial

(+) - livre

◀ Voltar



# Fim