Apontamentos

de

Optimização em Apoio à Decisão

Ano Lectivo de 2004/2005

Luís Nunes Vicente

Departamento de Matemática da F.C.T.U.C.

O curso de Optimização em Apoio à Decisão, com a duração de 22.5 horas, constitui uma das partes do módulo de Matemática (na Decisão) do Curso de Pós-Graduação em Informação (Geográfica) e Decisão do Departamento de Matemática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. Este curso de pós-graduação funciona em horário pós-laboral, para técnicos e quadros superiores a trabalhar fora do meio académico.

O curso foi organizado em torno dos seguintes tópicos:

1. Modelos de Divisão e Combinação de Recursos.
2. Conceitos Gerais em Programação Linear — Custos Reduzidos.
3. Dualidade em Programação Linear — Preços-Sombra.
4. Problemas de Fluxo em Redes.
5. Introdução à Programação Linear Multi-Objectivo.
6. Análise de Eficiência de Organizações (Metodologia DEA).
7. Modelos de Gestão de Projectos.
8. Modelos de Orçamento de Investimentos — Introdução à Programação Linear Inteira.
9. Formas de Modelação em Programação Linear Inteira.
10. Problemas de Afectação.
11. Modelos de Localização de Equipamentos e de Projecto de Redes.
12. Optimização de Rotas de Veículos.

Cada tópico corresponde a, sensivelmente, uma aula de 75–90 minutos. Os exemplos numéricos foram todos corridos com o *software* WinQSB. Os apontamentos não incluem as ilustrações em $\mathbb{R}^2$ feitas na aula, no quadro ou com o WinQSB.

Foi feito um esforço em reduzir a um mínimo o material sobre optimização. Essencialmente, procurou-se dar os conhecimentos básicos de optimização para modelar os problemas e interpretar e utilizar os resultados numéricos obtidos.

O curso foi complementado com dois seminários sobre casos práticos da utilização da optimização em apoio à decisão:

- Doutor Pedro Coimbra Martins (ISCAC, IPC), *Afectação de Trabalhadores a Escalas de Horários numa Estação de Tratamento de Correio.*
- Doutor António Pais Antunes (Dep. de Eng. Civil, FCTUC), *Modelos de Optimização Inteira para o Planeamento de Redes Educativas: O Caso de Condeixa-a-Nova.*

Coimbra, 20 de Janeiro de 2005, LNV.

Aula 1: Modelos de Divisão e Combinação de Recursos

Os modelos de divisão (*allocation*) e combinação (*blending*) de recursos formam alguns dos exemplos mais simples da programação linear.

Nos modelos de divisão de recursos, pretende-se dividir ou distribuir recursos por elementos em competição entre si. Os recursos podem ser os mais variados possíveis (financiamento, tempo, combustível, imobiliário, *etc.*).

Exemplo de um Modelo de Divisão de Recursos. Suponha que uma empresa tem 3 técnicos superiores a trabalhar em 4 projectos, que decorrerão nas duas próximas semanas. Cada técnico dispõe de 80 horas de trabalho para dividir pelos projectos. O tempo é o recurso a dividir. O número de horas necessário para concluir os projectos é dado por:

Projecto	1	2	3	4
Duração	70	50	85	35

A forma de contribuir para os projectos varia consoante os técnicos, de acordo com as suas qualificações e com a natureza do trabalho dos projectos. O gestor da empresa estima a capacidade de contribuição dos técnicos aos projectos, medida numa escala de 1 a 100, da seguinte forma:

	Projectos			
Técnicos	1	2	3	4
1	90	80	10	50
2	60	70	50	65
3	70	40	80	85

Seja t_{ij}, com $i = 1, 2, 3$ e $j = 1, 2, 3, 4$, o tempo que o técnico i dispenderá no projecto j. Neste modelo, pretende-se maximizar a capacidade total

$$90t_{11} + 80t_{12} + 10t_{13} + 50t_{14} + 60t_{21} + 70t_{22} + 50t_{23} + 65t_{24} + 70t_{31} + 40t_{32} + 80t_{33} + 85t_{34}.$$

A maximização desta capacidade total está restringida à duração de cada projecto:

$$\begin{aligned}
t_{11} + t_{21} + t_{31} &= 70,\\
t_{12} + t_{22} + t_{32} &= 50,\\
t_{13} + t_{23} + t_{33} &= 85,\\
t_{14} + t_{24} + t_{34} &= 35.
\end{aligned}$$

O programa linear que modela o problema inclui a não negatividade das variáveis t_{ij} e a garantia de que cada técnico trabalhará as 80 horas previstas:

$$\begin{array}{rl} \text{maximizar} & \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} t_{ij} \\ \text{sujeito a} & \sum_{i=1}^{3} t_{ij} \; = \; d_j, \quad j = 1,2,3,4, \\ & \sum_{j=1}^{4} t_{ij} \; = \; 80, \quad i = 1,2,3, \\ & t_{ij} \; \geq \; 0, \quad i = 1,2,3, \quad j = 1,2,3,4. \end{array}$$

em que d_j, $j = 1,2,3,4$, são os elementos da primeira tabela e c_{ij} é o elemento da linha i e da coluna j da segunda tabela.

Uma solução para este programa linear é a tabelada por:

	Projectos			
Técnicos	1	2	3	4
1	70	10	0	0
2	0	40	5	35
3	0	0	80	0

A capacidade total correspondente a esta solução é de 18825. O técnico 1, por exemplo, gastará 70 horas no projecto 1 e 10 horas no projecto 2, ignorando os outros dois.

Curiosamente, este problema pode ser visto como um problema de transportes (a ser apresentado mais à frente neste curso). Se substituíssemos as três restrições relativas ao número de horas de cada técnico por uma única restrição, da forma

$$\sum_{i=1}^{3} \sum_{j=1}^{4} t_{ij} \; = \; 240,$$

o problema deixaria de ser visto como um problema de transportes. A solução deste novo problema corresponde a:

	Projectos			
Técnicos	1	2	3	4
1	70	50	0	0
2	0	0	0	35
3	0	0	85	0

Nos problemas de combinação de recursos, estes últimos são partilhados de forma a verificar determinados requisitos da melhor forma possível. Neste tipo de problemas, os recursos são, geralmente, ingredientes, produtos ou materiais. Alguns dos problemas mais conhecidos desta classe são o problema da dieta e o problema da mistura em produção de materiais.

Exemplo de um Modelo de Combinação de Recursos. Imagine um negócio baseado na comercialização de um equipamento, com três tipos de funcionalidade (1, 2 e 3). O custo de comercialização de uma unidade do equipamento, varia consoante o seu tipo: 24 para o primeiro; 14 para o segundo; 8 para o terceiro.

Há duas classes de clientes a satisfazer (A e B). O cliente da classe A é menos exigente do que o cliente da classe B.

Sejam x_1, x_2 e x_3 o número de equipamentos dos tipos 1, 2 e 3 a comercializar.

Uma unidade do equipamento do tipo 1 pode ser partilhada por 4 clientes da classe A, satisfazendo as suas necessidades. Uma unidade do equipamento do tipo 2 satisfaz 3 clientes da classe A e uma unidade do equipamento do tipo 3 é partilhada por apenas um cliente desta classe. Há, pelo menos, 100 clientes da classe A a satisfazer, o que é traduzido por

$$4x_1 + 3x_2 + x_3 \ \geq \ 100.$$

A restrição correspondente aos clientes da classe B, em número não inferior a 65, é a seguinte:

$$2x_1 + x_2 + x_3 \ \geq \ 65.$$

O programa linear que modela este problema de combinação de recursos escreve-se na forma

$$\begin{aligned} \text{minimizar} \quad & 24x_1 + 14x_2 + 8x_3 \\ \text{sujeito a} \quad & 4x_1 + 3x_2 + x_3 \ \geq \ 100, \\ & 2x_1 + x_2 + x_3 \ \geq \ 65, \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

A solução consiste em comercializar 0 unidades do equipamento do tipo 1, 17.5 unidades do equipamento do tipo 2 e 47.5 unidades do equipamento do tipo 3, com um custo total de 625.

Este modelo coloca duas questões, relevantes em programação linear:

1. Se se pretender comercializar algumas unidades do equipamento de tipo 1, qual é o custo unitário dessa operação?

2. No caso de ser possível fornecer equipamento a mais clientes, qual é a classe de cliente à qual se deve dar primeiro resposta (no sentido de acarretar um custo por cliente inferior)? E qual é esse custo por cliente?

A primeira questão levantada no final do modelo de combinação de recursos, apresentado em cima, é respondida através da representação da solução obtida na forma de um *ponto básico admissível*. O *custo reduzido* (ou custo de oportunidade) associado à variável x_1 é o custo unitário procurado.

A resposta à segunda questão requer a formulação do *problema dual* associado ao programa linear original. Os valores da solução dual (*preços-sombra*) são os custos por cliente desejados.

Aula 2: Conceitos Gerais em Programação Linear — Custos Reduzidos

Um programa linear escrito na forma padrão (*standard*) escreve-se na forma

$$\begin{aligned} \text{minimizar} \quad & c^\top x \\ \text{sujeito a} \quad & Ax = b, \\ & x \geq 0, \end{aligned}$$

com $b \geq 0$. Nesta formulação, $x \in \mathbb{R}^n$ é o vector das incógnitas, $A \in \mathbb{R}^{m\times n}$ é a matriz das restrições e $b \in \mathbb{R}^m$ é o termo independente das restrições.

Por exemplo, o programa linear (com três variáveis e duas restrições)

$$\begin{aligned} \text{minimizar} \quad & 4x_1 - 5x_2 + 3x_3 \\ \text{sujeito a} \quad & 3x_1 - 2x_2 + 7x_3 = 7, \\ & 8x_1 + 6x_2 + 6x_3 = 5 \\ & x_1, x_2, x_3, \geq 0, \end{aligned}$$

encontra-se escrito na forma padrão. Em termos matriciais, temos que

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad c = \begin{bmatrix} 4 \\ -5 \\ 3 \end{bmatrix}, \quad A = \begin{bmatrix} 3 & -2 & 7 \\ 8 & 6 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 5 \end{bmatrix}.$$

Informações a reter sobre a forma padrão:

- Minimiza-se, em vez de se maximizar.
- As variáveis só podem tomar valores não negativos.
- As componentes do vector b são não negativas.

Note-se que todo o programa linear pode ser escrito na forma padrão.

Se o problema tiver uma função a ser maximizada, por exemplo,

$$\text{maximizar} \;\; 4x_1 - 3x_2 + 6x_3 \quad \left(= c^\top x\right),$$

o truque é multiplicar essa função por -1 e passar a

$$\text{minimizar} \quad -4x_1 + 3x_2 - 6x_3 \quad \left(= -c^\top x\right),$$

Uma restrição de desigualdade, do tipo

$$2x_1 + 7x_2 + 3x_3 \leq 10$$

é equivalente a

$$2x_1 + 7x_2 + 3x_3 + s_1 = 10 \quad \text{e} \quad s_1 \geq 0.$$

A variável adicional s_1 é designada por *variável de folga*.

Uma variável com limite inferior diferente de zero, $x_1 \geq 5$, pode ser substituída por $x_1' = x_1 - 5 \Leftrightarrow x_1 - x_1' = 5$ e $x_1' \geq 0$.

Finalmente, uma variável sem limites (*variável livre*) é substituída por um par de variáveis com valores não negativos: x_2 livre dá lugar a $x_2 = x_2' - x_2''$ com $x_1', x_2'' \geq 0$.

O conjunto

$$\{x \in \mathbb{R}^n : \ Ax = b, \ x \geq 0\}$$

chama-se a *região admissível* do programa linear (ou o conjunto dos pontos admissíveis). A função $c^\top x$ é designada por *função objectivo*.

A região admissível pode ser um conjunto vazio (programa linear inadmissível). Se for um conjunto não vazio, então podem acontecer três coisas:

- A solução (ou solução óptima) é única.
- Existem uma infinidade de soluções óptimas para as quais a função objectivo tem o mesmo valor (num conjunto limitado e fechado).
- O programa linear diz-se ilimitado e não existe solução óptima.

Um programa linear é um problema de optimização convexo (a função objectivo é uma *função convexa* e a região admissível é um *conjunto convexo*). Por este motivo, não podem existir minimizantes locais (ou relativos) com valores objectivos diferentes. Num problema de optimização convexo todo o minimizante local é global (ou absoluto).

Fazer exemplos com duas variáveis no quadro e com o *software*. Falar, informalmente, sobre pontos extremos, direcções de ilimitação e restrições activas.

Existem, essencialmente, duas classes de métodos para resolver programas lineares. No primeiro grupo estão o *método simplex* e as suas várias variantes (e, em particular, o *método dual-simplex*). Trata-se de métodos de *restrições activas*, que percorrem *pontos extremos* da região admissível. A outra classe de métodos é conhecida por métodos de pontos interiores. Nestes, a solução é alcançada pelo "interior" da região admissível. Os métodos de pontos interiores são conhecidos por serem numericamente eficientes para problemas de grandes dimensões e por atingirem uma complexidade de resolução *polinomial* (o método simplex atinge um desempenho *exponencial* em análise de pior instância). Para problemas de dimensão pequena ou média (até milhares de variáveis ou restrições), o método simplex revela-se satisfatório do ponto de vista da eficiência numérica. Este método mostra-se adequado em contextos práticos, por fornecer, de imediato ao utilizador, a informação de uma forma interpretável economicamente.

Para poder interpretar melhor o significado dos chamados *custos reduzidos* e dar uma resposta à primeira das questões levantadas no final da aula anterior, torna-se necessário abordar o conceito de *ponto básico admissível*. Sabe-se que qualquer programa linear (que não seja ilimitado ou inadmissível) *tem pelo menos uma solução óptima que é um ponto básico admissível*. Sabe-se, também, que um *ponto é extremo se e só se for um ponto básico admissível*.

O conceito de ponto básico admissível vai ser ilustrado para programas lineares escritos na forma padrão. A sua definição (rigorosa) é omitida. Se x for um ponto básico admissível então existe uma partição de x em *variáveis básicas* (em número m) e *variáveis não básicas* (em número $n-m$). Suponhamos que as básicas aparecem todas em primeiro lugar:

$$x \;=\; \begin{bmatrix} x_B \\ x_N \end{bmatrix}.$$

De forma análoga, tem-se que

$$A \;=\; \begin{bmatrix} B & N \end{bmatrix}.$$

A matriz B, de dimensões $m \times m$, é não singular (consequência da definição de ponto básico).

A partir desta partição e de $Ax = b$, escreve-se

$$Bx_B + Nx_N \;=\; b,$$

o que é equivalente a

$$x_B \;=\; B^{-1}b - B^{-1}Nx_N.$$

Como o ponto em causa é básico, $x_N = 0$. As variáveis não básicas tomam o valor zero na forma padrão. Se a região admissível fosse dada por $\{x \in \mathbb{R}^n : \; Ax = b, \; \ell \leq x \leq u\}$, com ℓ e u vectores em $\mathbb{R}^n$, então uma variável não básica x_i toma o valor ℓ_i ou u_i.

Como o ponto básico em consideração é admissível vem que $B^{-1}b \geq 0$. O valor da função objectivo num ponto básico admissível é dado por:

$$c^\top x \;=\; c_B^\top x_B + c_N^\top x_N \;=\; c_B^\top B^{-1}b + \left(c_N^\top - c_B^\top B^{-1}N\right) x_N.$$

Como $x_N = 0$, o valor objectivo é igual a $c_B^\top B^{-1}b$. No entanto, o vector

$$\hat{c}_N \;=\; c_N - N^\top (B^{-1})^\top c_B,$$

que é o transposto de $c_N^\top - c_B^\top B^{-1}N$, desempenha um papel relevante em diversas aplicações. As componentes deste vector são conhecidas por *custos reduzidos*.

- Um ponto básico admissível é uma solução do programa linear quando:

$$\hat{c}_N \;\geq\; 0.$$

- Um ponto básico admissível é uma solução única do programa linear quando:

$$\hat{c}_N \;>\; 0.$$

Se tentarmos aumentar o valor de uma váriavel não básica i (que tem o valor zero), esse aumento terá um custo unitário $(\hat{c}_N)_i$ no valor da função objectivo. É esta a interpretação económica do significado de um custo reduzido. No exemplo do modelo de combinação de recursos da Aula 1, como o custo reduzido associado à variável não básica x_1 é igual a 24, o custo a suportar por cada unidade do equipamento de tipo 1 a comercializar seria de 24.

- Um ponto básico admissível (ou não admissível) diz-se degenerado quando uma das componentes de $B^{-1}b$ é nula. Neste caso, há outras formas de representar *o mesmo ponto*, variando a partição em variáveis básicas e não básicas.

Ilustrar, geometricamente, o conceito de ponto básico admissível, não admissível e degenerado. Ilustrar a análise de sensibilidade.

Aula 3: Dualidade em Programação Linear — Preços-Sombra

Associado a um programa linear está outro programa linear designado por *programa linear dual* ou, simplesmente, por *dual*.

O dual de um programa linear admite uma interpretação económica relevante (para além de diversas propriedade teóricas e manipulações importantes).

Para introduzirmos o dual de um programa linear, vamos considerar que este está escrito na *forma canónica*:

$$\begin{aligned} \text{minimizar} \quad & c^\top x \\ \text{sujeito a} \quad & Ax \ \geq \ b, \\ & x \ \geq \ 0, \end{aligned}$$

Nesta formulação, $x \in \mathbb{R}^n$ é o vector das incógnitas, $A \in \mathbb{R}^{m\times n}$ é a matriz das restrições e $b \in \mathbb{R}^m$ é o termo independente das restrições. Há duas diferenças relativamente à forma padrão. Por um lado, não se exige que b tenha componentes não negativas. Por outro, as igualdades dão lugar a desigualdades do tipo $\geq$.

O dual de um programa linear, escrito na forma canónica, é dado por:

$$\begin{aligned} \text{maximizar} \quad & b^\top y \\ \text{sujeito a} \quad & A^\top y \ \leq \ c, \\ & y \ \geq \ 0, \end{aligned}$$

Também se chama canónica a esta forma de representar o dual.

Repare-se que, no programa linear original (*primal*) e no seu dual, os papéis dos vectores b e c aparecem trocados. Além disso, a matriz das restrições é a transposta uma da outra. O primal é um problema de minimização, enquanto o dual é de maximização.

Por exemplo, o dual de

$$\begin{aligned} \text{minimizar} \quad & 6x_1 + 2x_2 - x_3 + 2x_4 \\ \text{sujeito a} \quad & 4x_1 + 3x_2 - 2x_3 + 2x_4 \ \geq \ 10, \\ & 8x_1 + x_2 + 2x_3 + 4x_4 \ \geq \ 18, \\ & x_1, x_2, x_3, x_4 \ \geq \ 0, \end{aligned}$$

é

$$\begin{aligned} \text{maximizar} \quad & 10y_1 + 18y_2 \\ \text{sujeito a} \quad & 4y_1 + 8y_2 \ \leq \ 6, \\ & 3y_1 + y_2 \ \leq \ 2, \\ & -2y_1 + 2y_2 \ \leq \ -1, \\ & 2y_1 + 4y_2 \ \leq \ 2, \\ & y_1, y_2 \ \geq \ 0. \end{aligned}$$

Qualquer programa primal, que não esteja escrito na forma canónica, pode ser reformulado, equivalentemente, de modo a passar a estar escrito nesta forma. A título de exemplo, note-se que uma restrição de igualdade,

$$x_1 + 5x_2 \; = \; 6$$

é equivalente às duas seguintes restrições de desigualdade

$$x_1 + 5x_2 \; \geq \; 6 \quad \text{e} \quad -x_1 - 5x_2 \; \geq \; -6.$$

O outro tipo de técnicas necessárias para efectuar a conversão na forma canónica foram exemplicadas para o caso da forma padrão.

Assim, é possível deduzir todas as regras de passagem ao dual. Estas regras são aplicáveis a qualquer formulação primal. Basicamente, as regras são as seguintes:

- Restrições primais do tipo $\leq$ correspondem a variáveis duais não positivas.
  Restrições primais do tipo $=$ correspondem a variáveis duais livres.
- Váriáveis primais não positivas correspondem a restrições duais do tipo $\geq$.
  Váriáveis primais livres correspondem a restrições duais do tipo $=$.
- Maximização primal passa a minimização dual.

Aplicando estas regras, vemos que o dual do programa linear primal

$$\begin{array}{rl} \text{maximizar} & 6x_1 + x_2 + x_3 \\ \text{sujeito a} & 4x_1 + 3x_2 - 2x_3 \; = \; 1, \\ & 6x_1 - 2x_2 + 9x_3 \; \geq \; 9, \\ & 2x_1 + 3x_2 + 8x_3 \; \leq \; 5, \\ & x_1 \geq 0, \; x_2 \leq 0, \; x_3 \text{ livre} \end{array}$$

é o programa linear

$$\begin{array}{rl} \text{minimizar} & y_1 + 9y_2 + 5y_3 \\ \text{sujeito a} & 4y_1 + 6y_2 + 2y_3 \; \leq \; 6, \\ & 3y_1 - 2y_2 + 3y_3 \; \geq \; 1, \\ & -2y_1 + 9y_2 + 8y_3 \; = \; 1, \\ & y_1 \text{ livre}, \; y_2 \geq 0, \; y_3 \leq 0. \end{array}$$

Como outro exemplo, diga-se que o dual de um primal escrito na forma padrão é dado por

$$\begin{array}{rl} \text{maximizar} & b^\top y \\ \text{sujeito a} & A^\top y \; \leq \; c, \\ & y \text{ livre}. \end{array}$$

A teoria da dualidade diz-nos que:

- Dado um ponto admissível x para o primal (escrito na forma padrão) e um ponto admissível y para o seu dual, então $c^\top x \geq b^\top y$ (*dualidade fraca*).
- Se o primal é ilimitado então o dual é inadmissível (a sua região admissível é um conjunto vazio), e viceversa (consequências da dualidade fraca).
- Se um deles (primal ou dual) tem solução óptima o outro também e, nesse caso, os valores objectivos das soluções óptimas do primal e do dual coincidem (*dualidade forte*).

A dualidade forte permite uma interpretação económica dos valores da solução óptima do dual. Como o valor objectivo correspondente à solução óptima do primal (neste caso na forma padrão) se pode escrever na forma

$$b^\top y_* \quad = \; \cdots \; = \; b^\top \underbrace{(B^{-1})^\top c_B}_{\substack{\shortparallel \\ y_*}},$$

em que y_* representa a solução óptima do dual, vem que cada componente j de y_* corresponde ao custo/lucro unitário que se obteria aumentando/diminuindo o recurso ou requisito b_j. As componentes de y_* são conhecidas, assim, por preços-sombra.

Os preços-sombra associados às duas restrições do exemplo do modelo de combinação de recursos da Aula 1 são 3 e 5, respectivamente. Assim sendo, no caso de ser possível fornecer equipamento a mais clientes, dever-se-ia dar primeiro resposta ao cliente da classe A (com um custo por cliente igual a 3).

O dual ocupa um papel relevante no aproveitamento da estrutura do problema, tendo em vista a eficiência da resolução numérica. Quando o número de restrições é muito superior ao número de variáveis, torna-se preferível resolver o programa dual, cujo número de restrições vai ser inferior ao número de variáveis. Na base desta observação está o facto da dimensão das matrizes dos sistemas a resolver pelo método simplex para determinar pontos básicos estar relacionado com o número de restrições. Nestes casos, o método simplex dá lugar ao método dual-simplex.

Uma outra propriedade importante, em contextos práticos, é a chamada *complementaridade das variáveis de folga*. Suponhamos o programa primal escrito na forma padrão. Esta propriedade diz, essencialmente, que se uma variável primal óptima é positiva então a correspondente desigualdade dual é activa na solução dual (e a correspondente variável de folga associada a esta restrição dual é nula).

Aula 4: Problemas de Fluxo em Redes

Os problemas de *fluxo de custo mínimo* em redes aparecem ligados a situações relacionadas com o transporte ou distribuição de bens e mercadorias entre diversos locais.

Um outro tipo de problemas de fluxo em redes, os de *fluxo máximo*, ocorre frequentemente como subproblema em operações complexas. É possível, por exemplo, modelar planos de evacuação de segurança através de problemas de fluxo máximo.

Um dos casos mais simples de problemas de fluxo de custo mínimo em redes e, consequentemente, de programação linear é o *problema de transportes*.

Suponhamos que uma determinada mercadoria vai ser transportada de um conjunto de locais onde se encontra armazenada (origens) para um outro conjunto de locais onde a mercadoria não existe e é procurada (destinos).

Sejam

$$s_i \stackrel{\text{def}}{=} \text{quantidade armazenada na origem } i,$$

e

$$d_j \stackrel{\text{def}}{=} \text{quantidade procurada no destino } j.$$

O objectivo é transportar toda a mercadoria das origens para os destinos, com um custo mínimo. Para esse efeito, seja

$$c_{ij} \stackrel{\text{def}}{=} \text{custo de transporte de uma unidade de mercadoria de } i \text{ para } j.$$

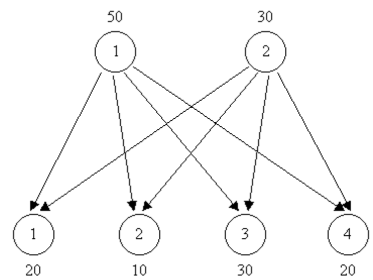
Designem-se por I e J, respectivamente, os conjuntos das origens e dos destinos. Designando por x_{ij} o número de unidades de mercadoria a enviar de i para j, o programa linear que modela o problema em causa é formulado na forma:

$$\begin{array}{rl} \text{minimizar} & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \\ \text{sujeito a} & \sum_{j\in J} x_{ij} \;=\; s_i, \quad i\in I, \\ & \sum_{i\in I} x_{ij} \;=\; d_j, \quad j \in J, \\ & x_{ij} \;\geq\; 0, \quad i\in I, \quad j\in J. \end{array}$$

Exemplo de um Problema de Transportes. Suponhamos que em duas origens está armazenada uma mercadoria, pronta a ser enviada para quatro destinos. Os custos unitários de transporte das origens para os destinos são dados por:

	Destinos			
Origens	1	2	3	4
1	5	4	8	9
2	10	1	7	1

As quantidades armazenadas nas origens são, respectivamente, 50 e 30, enquanto as quantidades procuradas nos destinos são iguais a, respectivamente, 20, 10, 30 e 20. A figura seguinte retrata o grafo orientado deste problema de transportes.



O programa linear correspondente a este problema de transportes é formulado na forma:

$$
\begin{aligned}
\text{minimizar} \quad & 5x_{11} + 4x_{12} + 8x_{13} + 9x_{14} + 10x_{21} + x_{22} + 7x_{23} + x_{24} \\
\text{sujeito a} \quad & x_{11} + x_{12} + x_{13} + x_{14} \ = \ 50, \\
& x_{21} + x_{22} + x_{23} + x_{24} \ = \ 30, \\
& x_{11} + x_{21} \ = \ 20, \\
& x_{12} + x_{22} \ = \ 10, \\
& x_{13} + x_{23} \ = \ 30, \\
& x_{14} + x_{24} \ = \ 20, \\
& x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24} \ \geq \ 0.
\end{aligned}
$$

Uma solução para este problema é dada por

$$x_{11} \ = \ 20, \quad x_{13} \ = \ 30, \quad x_{22} \ = \ 10, \quad x_{24} \ = \ 20$$

(com as restantes variáveis a tomar o valor zero).

Como dissemos anteriormente, um problema de transportes é um caso particular de um problema de fluxo de custo mínimo, que passamos a descrever.

Consideremos uma rede orientada (grafo orientado com informação suplementar em nodos e arcos), com um conjunto de nodos N e um conjunto de arcos $A \subset N \times N$. Os nodos podem ser de três tipos:

- Nodos de origem: onde a mercadoria está armazenada para ser fornecida.
- Nodos de destino: onde a mercadoria é procurada.

- Nodos de entreposto: por onde toda a mercadoria que entra, sai necessariamente.

Associados aos arcos estão custos de transporte: c_{ij} é o custo de transporte do nodo i para o nodo j, quando $(i,j) \in A$. Associado a todo o nodo está uma quantidade de procura, que no caso dos nodos de origem é negativa (procura negativa significa provisão) e no caso dos nodos de entreposto é nula.

As variáveis não negativas do problema (x_{ij} para todo o $(i,j) \in A$) correspondem ao fluxo de mercadoria que passa em cada arco. Associado ainda a cada arco, pode estar um limite superior de fluxo: $x_{ij} \leq u_{ij} < +\infty$. Chama-se a u_{ij} a capacidade do arco $(i,j) \in A$.

A formulação geral de um problema de fluxo de custo mínimo é dada pelo programa linear:

$$\begin{array}{rl} \text{minimizar} & \sum_{(i,j)\in A} c_{ij}x_{ij} \\ \text{sujeito a} & \sum_{(i,k)\in A} x_{ik} - \sum_{(k,j)\in A} x_{kj} \;=\; b_k, \quad \text{para todo o } k \in N, \\ & 0 \;\leq\; x_{ij} \;\leq\; u_{ij} \quad \text{para todo o } (i,j) \in A. \end{array}$$

Nos nodos de entreposto, tem-se que $b_k = 0$, e as restrições

$$\sum_{(i,k)\in A} x_{ik} - \sum_{(k,j)\in A} x_{kj} \;=\; 0$$

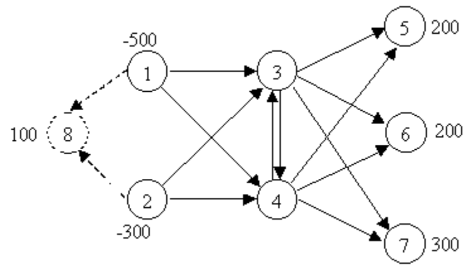
traduzem a conservação de fluxo nesses nodos (tudo o que entra sai).

Exemplo de um Problema de Fluxo de Custo Mínimo. Um exemplo de um problema de fluxo de custo mínimo com 2 origens (nodos 1 e 2), 3 destinos (nodos 5, 6 e 7) e 2 nodos de entreposto (3 e 4) é o seguinte:

$$\begin{array}{rl} \text{minimizar} & 5x_{13} + 8x_{14} + 8x_{23} + 5x_{24} + 10x_{35} + 10x_{36} + 10x_{37} + 10x_{45} + 10x_{46} + 10x_{47} \\ \text{sujeito a} & -x_{13} - x_{14} - x_{18} \;=\; -500, \\ & -x_{23} - x_{24} - x_{28} \;=\; -300, \\ & +x_{13} + x_{23} + x_{43} - x_{34} - x_{35} - x_{36} - x_{37} \;=\; 0, \\ & +x_{14} + x_{24} + x_{34} - x_{43} - x_{45} - x_{46} - x_{47} \;=\; 0, \\ & +x_{35} + x_{45} \;=\; 200, \\ & +x_{36} + x_{46} \;=\; 200, \\ & +x_{37} + x_{47} \;=\; 300, \\ & +x_{18} + x_{28} \;=\; 100, \\ & x_{34} \;\leq\; 100, \qquad x_{43} \;\leq\; 100, \\ & x_{ij} \;\geq\; 0 \quad \text{para todo o } (i,j) \in A. \end{array}$$

Apenas dois arcos apresentam restrições de capacidade de fluxo (os arcos $(3,4)$ e $(4,3)$). Os dois nodos de origem têm procuras dadas por -500 e -300 (ou seja, provisões 500 e

300). A procura nos três nodos de destino é 200, 200 e 300, respectivamente. Como a quantidade abastecida supera a quantidade procurada, houve necessidade de incluir um nodo artificial de destino (o nodo 8), ligado às origens com custo zero. A figura seguinte retrata o grafo orientado correspondente a este problema.



O valor óptimo deste programa linear é 10500. Uma solução é dada por:

$$x_{13} = 400, \quad x_{24} = 300, \quad x_{43} = 100, \quad x_{36} = 200, \quad x_{37} = 300, \quad x_{45} = 200, \quad x_{18} = 100$$

(com as restantes variáveis a tomar o valor zero).

Esta classe de problemas de fluxo de custo mínimo é conhecida por problemas de transbordo (*transshipment*), por estar associada ao transporte de mercadorias de meios navais para ferroviários ou viceversa.

A formulação de fluxo de custo mínimo para o problema de transportes é dada por

$$\begin{array}{rl} \text{minimizar} & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} \\ \text{sujeito a} & -\sum_{j\in J} x_{ij} \;=\; -s_i, \quad i\in I, \\ & \sum_{i\in I} x_{ij} \;=\; d_j, \quad j\in J, \\ & x_{ij} \;\geq\; 0, \quad i\in I, \quad j\in J. \end{array}$$

Não há nodos de entreposto nos problemas de transportes.

Os problemas de fluxo de custo mínimo são inadmissíveis (região admissível vazia) se a soma das quantidades procuradas não coincidir com a soma das quantidades a fornecer, ou seja, se

$$\sum_{k \in N} b_k \neq 0.$$

Esta condição, nos problemas de transportes, assume a forma

$$\sum_{i \in I} s_i \neq \sum_{j \in J} d_j.$$

Uma forma de contornar este problema é introduzir um nodo artificial, respectivamente de origem ou de destino, ligado por arcos de custo unitário nulo a todos os nodos de destino ou de origem (respectivamente) do problema em causa. No exemplo anterior, aplicou-se este procedimento incluindo um nodo de destino artificial (com procura igual a 100), ligado aos dois nodos de origem a custo zero.

Os problemas considerados até ao momento envolvem apenas o transporte de um tipo mercadoria. Não é difícil, no entanto, modelar a situação em que existe mais do que um tipo de mercadoria.

Nos problemas de fluxo máximo, há apenas um nodo de origem e um nodo de destino. Todos os restantes nodos são entrepostos que conservam o fluxo de passagem. Pretende-se enviar o máximo fluxo (pessoas no caso de um plano de evacuação) da origem para o destino. Nestes problemas, as restrições de capacidade de fluxo nos arcos desempenham um papel mais relevante. A formulação geral de um problema de fluxo máximo é dada por:

$$\begin{array}{rl}
\text{maximizar} & \sum_{(i,j) \in A} x_{ij} \\
\text{sujeito a} & \sum_{(i,k) \in A} x_{ik} - \sum_{(k,j) \in A} x_{kj} = 0, \quad \text{para todo o entreposto } k, \\
 & 0 \leq x_{ij} \leq u_{ij} \quad \text{para todo o } (i,j) \in A.
\end{array}$$

As matrizes das restrições destes problemas de fluxo em redes (que descrevem a variação de fluxo em todos os nodos) são designadas por matrizes de incidência nodo-arco. Estas matrizes gozam de uma propriedade especial, a *unimodularidade total*. Uma matriz é totalmente unimodular se toda a sua submatriz quadrada tiver determinante igual a $+1$, 0 ou -1. É por este motivo que, se as componentes do termo independente forem inteiras, qualquer ponto básico (e em particular uma solução óptima que seja básica) tem componentes também inteiras. Observámos este fenómeno ao longo desta aula e, também, na primeira versão do problema de divisão de recursos apresentado anteriormente (por este ser um caso particular do problema de transportes).

Aula 5: Introdução à Programação Linear Multi-Objectivo

É frequente encontrar, em problemas de índole prática, mais do que uma função objectivo a optimizar, para a mesma região admissível. Esta ocorrência altera o conceito clássico de solução ou solução óptima, que deixa de fazer sentido. Entre os problemas mais simples de optimização multi-objectivo, mas com um campo de aplicações muito vasto, encontra-se a programação linear multi-objectivo.

A programação linear multi-objectivo é motivada, neste curso, através do problema de gestão de *cash-flows*, apresentado de seguida, que pode ser visto à luz dos modelos de divisão/combinação de recursos. Neste problema, serão definidos dois objectivos a optimizar simultaneamente (e, como se verá, conflituosamente).

Exemplo de um Problema de Gestão de Cash-Flows. Uma empresa tem dois projectos em carteira (projecto A e projecto B), com um tempo de vida de três anos. Estima-se, para estes dois projectos, que a evolução de *cash-flows* (em dezenas de milhares de euros) seja a seguinte:

	Anos		
Projectos	2005	2006	2007
A	-0.5	0.4	0.6
B	-1.0	1.25	1.75

O projecto A, por exemplo, necessitará de um investimento de 0.5 em 2005, dando um retorno de 0.4 em 2006 e de 0.6 em 2007 (em dezenas de milhares de euros).

A empresa não dispõe de mais de uma dezena de milhares de euros para investir nos projectos em 2005. Sejam x_1 e x_2 as proporcões (de 0 a 1) de investimento nos projectos A e B, respectivamente. Quando $x_2 = 1$, a impresa investe a 100% no projecto B. Se $x_2 = 0.4$ então o investimento neste projecto passa a ser somente de 40%. Por razões técnicas, a empresa está impedida de investir mais de 75% no segundo projecto. A região admissível que delimita as escolhas possíveis é, assim, definida por

$$0.5x_1 + x_2 \le 1, \quad 0 \le x_1 \le 1, \quad 0 \le x_2 \le 0.75.$$

A maior parte dos gestores da empresa valoriza o retorno em ambos os projectos (nos dois anos subsequentes), indiferentemente. Neste sentido, a função a maximizar seria

$$x_1 + 3x_2.$$

(Note que $1 = 0.4 + 0.6$ e que $3 = 1.25 + 1.75$.) Teríamos, neste cenário, o seguinte programa linear a resolver:

$$\begin{aligned} \text{maximizar} \quad & x_1 + 3x_2 \\ \text{sujeito a} \quad & 0.5x_1 + x_2 \ \leq \ 1, \\ & 0 \ \leq \ x_1 \ \leq \ 1, \\ & 0 \ \leq \ x_2 \leq \ 0.75. \end{aligned}$$

A solução deste programa linear é dada por $x_1 = 0.5$ e $x_2 = 0.75$.

Existe, porém, uma parte da administração que privilegia, por razões estratégicas, o retorno do projecto A. O programa linear a considerar, neste caso, seria dado por

$$\begin{aligned} \text{maximizar} \quad & x_1 \\ \text{sujeito a} \quad & 0.5x_1 + x_2 \ \leq \ 1, \\ & 0 \ \leq \ x_1 \ \leq \ 1, \\ & 0 \ \leq \ x_2 \leq \ 0.75. \end{aligned}$$

Este programa não tem solução única. São soluções deste programa linear os pontos no segmento de recta de extremidades $(1, 0)$ e $(1, 0.5)$.

Resolver, geometricamente, no quadro ou com o *software*, os dois programas lineares.

Se tomarmos ambas as funções objectivo, obtemos o programa linear multi-objectivo (neste caso bi-objectivo):

$$\begin{aligned} \text{maximizar} \quad & x_1 + 3x_2 \\ \text{maximizar} \quad & x_1 \\ \text{sujeito a} \quad & 0.5x_1 + x_2 \ \leq \ 1, \\ & 0 \ \leq \ x_1 \ \leq \ 1, \\ & 0 \ \leq \ x_2 \leq \ 0.75. \end{aligned}$$

Seja S a sua região admissível. Os *pontos eficientes* deste programa linear bi-objectivo são os pontos de S que não são *dominados* por nenhum outro, ou seja, são todos os pontos (x_1, x_2) em S para os quais não existe $(y_1, y_2) \in S$ tal que

$$y_1 + 3y_2 \ > \ x_1 + 3x_2 \quad \text{e} \quad y_1 \ > \ x_1.$$

Os pontos eficientes deste programa linear bi-objectivo são os pertencentes ao conjunto

$$\begin{aligned} F \ = \ & \{(x_1, x_2) \in \mathbb{R}^2 : \ 0.5x_1 + x_2 = 1, \ 0.5 \leq x_1 \leq 1\} \\ & \cup \ \{(x_1, x_2) \in \mathbb{R}^2 : \ x_1 = 1, \ 0 \leq x_2 \leq 0.5\} \,. \end{aligned}$$

Dado um ponto em F, não existe nenhum outro ponto em S melhor do que ele em ambos os objectivos simultaneamente.

Os *pontos estritamente eficientes* deste programa linear bi-objectivo são os pontos (x_1, x_2) em S para os quais não existe $(y_1, y_2) \in S$ tal que

$$y_1 + 3y_2 \;\geq\; x_1 + 3x_2 \quad \text{e} \quad y_1 \;\geq\; x_1 \quad \text{(com pelo menos uma desigualdade } >\text{).}$$

Os pontos estritamente eficientes deste programa linear bi-objectivo são os que estão em

$$F_e \;=\; \left\{(x_1, x_2) \in \mathbb{R}^2 : \; 0.5x_1 + x_2 = 1, \; 0.5 \leq x_1 \leq 1\right\}.$$

Ilustrar, geometricamente, no quadro ou com o *software*, as fronteiras de eficiência e de eficiência estrita.

De uma forma geral, um programa linear multi-objectivo (com q objectivos) pode ser escrito na forma

$$\begin{array}{ll} \text{maximizar} & (c^1)^\top x \\ \vdots & \vdots \\ \text{maximizar} & (c^q)^\top x \\ \text{sujeito a} & x \in S, \end{array}$$

em que S designa a região admissível. A ordenação das funções objectivo é irrelevante para a definição dos conjuntos F e F_e. Se o problema incluísse minimizações, estas poderiam ser passadas a maximizações.

Os *pontos eficientes* de um programa linear multi-objectivo são os pontos de S que não são *dominados* por nenhum outro, ou seja, são todos os pontos x em S para os quais não existe $y \in S$ tal que

$$(c^i)^\top y \;>\; (c^i)^\top x, \quad i = 1, \ldots, q.$$

Ao conjunto F contendo todos os pontos eficientes dá-se o nome de *fronteira de eficiência.*

Os *pontos estritamente eficientes* de um programa linear multi-objectivo são os pontos x em S para os quais não existe $y \in S$ tal que

$$(c^i)^\top y \;\geq\; (c^i)^\top x, \quad i = 1, \ldots, q \qquad \text{(com pelo menos uma desigualdade } >\text{).}$$

O conjunto F_e dos pontos estritamente eficientes é designado por fronteira de eficiência estrita.

Três propriedades básicas da programação linear multi-objectivo são as seguintes:

- $F_e \subset F$.
- Todas as soluções (únicas ou não) dos programas lineares individuais estão em F.
- Pelo menos uma das soluções de cada um dos programas lineares individuais está em F_e.

Um ponto em F_e pode ser obtido aplicando o *método lexicográfico*, em que se maximizam as funções objectivo sucessivamente. Primeiro maximiza-se a primeira função objectivo. Suponhamos que o conjunto das soluções óptimas obtido não era singular. A seguir maximiza-se a segunda função objectivo neste conjunto de soluções óptimas, e assim sucessivamente.

Um outro processo de obter pontos em F, ou em F_e, é através da ponderação das funções objectivo. Para o efeito, considere o programa linear (dependente dos pesos $\lambda_1, \ldots, \lambda_q$)

$$\begin{array}{ll} \text{maximizar} & \sum_{i=1}^{q} \lambda_i \, (c^i)^\top x \\ \text{sujeito a} & x \in S, \end{array}$$

que designaremos por $PL(\lambda)$.

Para descrever F, tomam-se todos os pesos em

$$\Lambda \;=\; \left\{ \lambda \in \mathbb{R}^q : \; \sum_{i=1}^{q} \lambda_i \;=\; 1, \quad \lambda_i \;\geq\; 0, \; i = 1, \ldots, q \right\}.$$

De facto, a fronteira de eficiência pode ser caracterizada por

$$F \;=\; \{ x \in \mathbb{R}^n : \; x \text{ é solução de } PL(\lambda) \text{ para } \lambda \in \Lambda \}.$$

A fronteira de eficiência estrita pode ser descrita por

$$F_e \;=\; \{ x \in \mathbb{R}^n : \; x \text{ é solução de } PL(\lambda) \text{ para } \lambda \in \Lambda_e \},$$

em que

$$\Lambda_e \;=\; \left\{ \lambda \in \mathbb{R}^q : \; \sum_{i=1}^{q} \lambda_i \;=\; 1, \quad \lambda_i \;>\; 0, \; i = 1, \ldots, q \right\}.$$

Assim sendo, atribuindo valores apropriados aos pesos λ_i, determinam-se pontos na fronteira de eficiência ou na fronteira de eficiência estrita. Por exemplo, no caso do exemplo anterior, se escolhessemos

$$\lambda_1 \;=\; 0.25 \quad \text{e} \quad \lambda_2 \;=\; 0.75,$$

obter-se-ia o programa linear $PL(0.25, 0.75)$

$$\begin{array}{rl} \text{maximizar} & 0.25(x_1 + 3x_2) + 0.75x_1 \;=\; x_1 + 0.75x_2 \\ \text{sujeito a} & 0.5x_1 + x_2 \;\leq\; 1, \\ & 0 \;\leq\; x_1 \;\leq\; 1, \\ & 0 \;\leq\; x_2 \leq\; 0.75. \end{array}$$

A solução deste programa linear é o ponto $(1, 0.5)$.

Se a escolha fosse

$$\lambda_1 = 0.75 \quad \text{e} \quad \lambda_2 = 0.25,$$

obter-se-ia o programa linear $PL(0.75, 0.25)$

$$\begin{aligned} \text{maximizar} \quad & 0.75(x_1 + 3x_2) + 0.25x_1 = x_1 + 2.25x_2 \\ \text{sujeito a} \quad & 0.5x_1 + x_2 \leq 1, \\ & 0 \leq x_1 \leq 1, \\ & 0 \leq x_2 \leq 0.75. \end{aligned}$$

A solução deste programa linear é o ponto $(0.5, 0.75)$.

Com a escolha $\lambda_1 = 2/3$ e $\lambda_2 = 1/3$, obter-se-ia o resto da fronteira de eficiência estrita, ou seja, o segmento de recta que os pontos $(1, 0.5)$ e $(0.5, 0.75)$.

Somente a escolha $\lambda_1 = 0$ e $\lambda_2 = 1$ permitiria determinar a parte da fronteira de eficiência que não está contida na fronteira de eficiência estrita.

A escolha dos pesos ou factores de ponderação está associada à preferência dos agentes de decisão. É frequente ser mais simples ao agente de decisão especificar metas a atingir (em termos quantitativos) para cada um dos objectivos do que expressar, em factores de ponderação, as suas preferências.

Uma técnica relacionada com a optimização multi-objectivo é a optimização por metas (*goal programming*), em que a optimização dos objectivos é substituída pela tentativa de satisfazer determinadas metas escolhidas para os mesmos. No caso de um programa linear bi-objectivo,

$$\begin{aligned} \text{maximizar} \quad & (c^1)^\top x \\ \text{maximizar} \quad & (c^2)^\top x \\ \text{sujeito a} \quad & x \in S, \end{aligned}$$

as metas (m_1 e m_2) seriam satisfeitas se conseguíssemos encontrar pontos x que verificassem

$$\begin{aligned} (c^1)^\top x &\geq m_1, \\ (c^2)^\top x &\geq m_2, \\ x &\in S. \end{aligned}$$

Como esta nova região admissível pode ser um conjunto vazio, a programação por metas consiste, neste caso bi-objectivo, na resolução do programa linear

$$\begin{aligned} \text{minimizar} \quad & d_1 + d_2 \\ \text{sujeito a} \quad & (c^1)^\top x + d_1 \geq m_1, \\ & (c^2)^\top x + d_2 \geq m_2, \\ & x \in S, \\ & d_1, d_2 \geq 0. \end{aligned}$$

As variáveis d_1 e d_2 são conhecidas por *variáveis de ineficiência*. A sua função é permitir uma violação das metas. O problema linear acima descrito tem por objectivo tornar esta violação o mais pequena possível.

Aula 6: Análise de Eficiência de Organizações (Metodologia DEA)

A metodologia *Data Envelopment Analysis* (DEA) tem por objectivo analisar a eficiência relativa do desempenho de unidades ou organizações.

Esta análise de eficiência aplica-se quando o desempenho de cada unidade foi medido, quantitativamente, por um conjunto de *inputs* (por exemplo, investimento, financiamento, recursos humanos, recursos materiais) e um conjunto de *outputs* (por exemplo, lucro, produção, atendimento, formação).

A metodologia DEA constrói uma medida simples para relacionar a soma ponderada dos *inputs* com a soma ponderada dos *outputs*, detectando a eficiência ou ineficiência de uma unidade relativamente às outras. Os pesos ou factores de ponderação são calculados pela própria metodologia e não estão sujeitos a critérios de escolha subjectivos.

O número de unidades e o número dos seus *inputs* e *outputs* podem ser quaisquer. Uma outra vantagem desta metodologia é admitir indicadores de *inputs* e de *outputs* dados em medições diferentes.

Exemplo de um Modelo de Análise de Eficiência de Unidades. Pretende-se analisar a eficiência relativa de cinco unidades de cirurgia localizadas em diferentes hospitais. Em cada unidade, conhece-se o número de cirurgiões e o financiamento em medicação (medido em dezenas de milhares de euros), constituindo estas duas quantidades os *inputs* em causa. Como *outputs* de cada unidade, são disponibilizados o número de cirurgias, a percentagem de ocupação de camas e o número de consultas externas, num dado período de tempo. Os dados são os seguintes:

Unidade 1	40	100	100	6	5
Unidade 2	30	85	110	7	10
Unidade 3	25	80	70	4	8
Unidade 4	32	95	20	5	5
Unidade 5	28	50	60	6	5

As três primeiras colunas de dados referem-se aos *outputs* e as duas últimas aos *inputs*.

Uma medida de eficiência da unidade 1 é definida através do quociente entre uma soma ponderada do seu *output* e uma soma ponderada do seu *input*, ou seja, através de

$$\frac{u_1 40 + u_2 100 + u_3 100}{v_1 6 + v_2 5},$$

para uma determinada escolha dos pesos ou factores de ponderação u_1, u_2 e u_3 (relativos aos *outputs*) e v_1 e v_2 (relativos aos *inputs*). A atribuição de valores aos factores de ponderação pode colidir com a dificuldade prática em valorizar subjectivamente os vários desempenhos.

A ideia central da metodologia DEA passa por determinar um conjunto de pesos que maximize a eficiência de uma unidade relativamente às restantes. Deste modo, é possível atribuir pesos de forma diferente, de unidade para unidade, aos vários *inputs* e *outputs*. Toda a eficiência é medida no intervalo $[0, 1]$ para poder ser facilmente interpretada em percentagem.

No caso da unidade 1, os pesos resultariam da resolução do problema de optimização

$$\begin{aligned}
\text{maximizar} \quad & \tfrac{u_1 40 + u_2 100 + u_3 100}{v_1 6 + v_2 5} \\
\text{sujeito a} \quad & \tfrac{u_1 40 + u_2 100 + u_3 100}{v_1 6 + v_2 5} \leq 1, \\
& \tfrac{u_1 30 + u_2 85 + u_3 110}{v_1 7 + v_2 10} \leq 1, \\
& \tfrac{u_1 25 + u_2 80 + u_3 70}{v_1 4 + v_2 8} \leq 1, \\
& \tfrac{u_1 32 + u_2 95 + u_3 20}{v_1 5 + v_2 5} \leq 1, \\
& \tfrac{u_1 28 + u_2 50 + u_3 60}{v_1 6 + v_2 5} \leq 1. \\
& u_1, u_2, u_3, v_1, v_2 > 0.
\end{aligned}$$

A função objectivo deste problema é *fraccionária* (trata-se de um problema de programação fraccionária). No entanto, assumindo que os pesos não podem tomar o valor zero, é possível reescrever este problema como um programa linear (com cinco variáveis e seis restrições)

$$\begin{aligned}
\text{maximizar} \quad & u_1 40 + u_2 100 + u_3 100 \\
\text{sujeito a} \quad & v_1 6 + v_2 5 = 1, \\
& u_1 40 + u_2 100 + u_3 100 \leq 1, \\
& u_1 30 + u_2 85 + u_3 110 - v_1 7 - v_2 10 \leq 0, \\
& u_1 25 + u_2 80 + u_3 70 - v_1 4 - v_2 8 \leq 0, \\
& u_1 32 + u_2 95 + u_3 20 - v_1 5 - v_2 5 \leq 0, \\
& u_1 28 + u_2 50 + u_3 60 - v_1 6 - v_2 5 \leq 0, \\
& u_1, u_2, u_3, v_1, v_2 \geq \epsilon,
\end{aligned}$$

em que ϵ é um número pequeno (por exemplo, $\epsilon = 10^{-8}$). Este programa linear tem valor óptimo 1. Uma solução corresponde aos valores das varíaveis dados por:

u_1^*	u_2^*	u_3^*	v_1^*	v_2^*
0.0190	0.0024	ϵ	0.1667	ϵ

O facto do valor óptimo ser 1 significa que a unidade 1 é eficiente relativamente às restantes. Os valores das variáveis de folga das restrições de desigualdade (correspondentes às cinco unidades) foram os seguintes:

s_1^*	s_2^*	s_3^*	s_4^*	s_5^*
0	0.3918	0	0	0.3477

É possível concluir que as unidades 3 e 4 atingiram uma eficiência máxima (1) para a escolha dos factores de ponderação que tornaram a unidade 1 o mais eficiente relativamente a todas as outras. Desta forma, as unidades 3 e 4 podem ser consideradas como unidades de referência alternativa da unidade 1.

No caso da unidade 2, a sua eficiência relativa seria determinada resolvendo o programa linear

$$\begin{aligned}
\text{maximizar} \quad & u_1 30 + u_2 85 + u_3 110 \\
\text{sujeito a} \quad & v_1 7 + v_2 10 \; = \; 1, \\
& u_1 40 + u_2 100 + u_3 100 - v_1 6 - v_2 5 \; \leq \; 0, \\
& u_1 30 + u_2 85 + u_3 110 \; \leq \; 1, \\
& u_1 25 + u_2 80 + u_3 70 - v_1 4 - v_2 8 \; \leq \; 0, \\
& u_1 32 + u_2 95 + u_3 20 - v_1 5 - v_2 5 \; \leq \; 0, \\
& u_1 28 + u_2 50 + u_3 60 - v_1 6 - v_2 5 \; \leq \; 0, \\
& u_1, u_2, u_3, v_1, v_2 \; \geq \; \epsilon,
\end{aligned}$$

que tem a mesma estrutura do anterior (com cinco variáveis e seis restrições). A solução obtida tem o valor objectivo 0.9194, o que mostra que a unidade 2 não é eficiente relativamente às restantes ($0.9194 < 1$). Os valores obtidos para as variáveis de folga das restrições de desigualdade são dados por:

s_1^*	s_2^*	s_3^*	s_4^*	s_5^*
0	0.0806	0	0.5343	0.3343

As unidades 1 e 3 constituem, assim, referência para a unidade 2. A solução obtida foi a seguinte:

u_1^*	u_2^*	u_3^*	v_1^*	v_2^*
ϵ	ϵ	0.0084	0.1343	0.0060

O facto de $u_1^* = u_2^* = \epsilon$ significa que a unidade 2 não atingiu o melhor desempenho possível nos dois primeiros indicadores de *output*. Seria possível com 91.94% dos recursos de *input*, para os mesmos indicadores de *output*, tornar esta unidade eficiente relativamente às outras. Em alternativa, como $1/0.9194 \simeq 1.0877$, seria possível fazer com que esta unidade ficasse eficiente relativamente às outras se se conseguisse melhorar em 8.77% os indicadores de *output* para os mesmos níveis de *input*.

E assim se procederia até à unidade 5, calculando a eficiência relativa de todas as unidades. A tabela apresentada a seguir mostra o grau de eficiência relativa, bem como as unidades de referência obtidas para as cinco unidades.

Unidades	Eficiência	Referências
1	100%	3,4 (alternativas)
2	91.94%	1,3
3	100%	1,4 (alternativas)
4	100%	1,3 (alternativas)
5	70%	1

A flexibilidade desta metodologia pode tornar relativamente eficiente (= 100%) uma unidade ao calcular um conjunto de pesos óptimos que, na prática, podem revelar algum grau de artificialidade. Por outro lado, quando uma unidade é rotulada como relativamente ineficiente (< 100%), existe a garantia de que nem a escolha dos factores de ponderação mais favorável possível conseguiu evitar essa rotulação.

Considere-se, agora, um cenário mais geral em que existem n unidades, p indicadores de *input* e q indicadores de *output*. Seja x_{sj} o valor do *input* s da unidade j. Seja y_{rj} o valor do *output* r da unidade j.

O programa linear que determina a eficiência da unidade $\ell \in \{1, \ldots, n\}$ relativamente às demais é formulado na forma:

$$\begin{array}{rl} \text{maximizar} & \sum_{r=1}^{q} y_{r\ell} u_r \\ \text{sujeito a} & \sum_{s=1}^{p} x_{s\ell} v_s \ = \ 1, \\ & \sum_{r=1}^{q} y_{rj} u_r - \sum_{s=1}^{p} x_{sj} v_s \ \leq \ 0, \quad j = 1, \ldots, n, \\ & u_1, \ldots, u_q, v_1, \ldots, v_p \ \geq \ \epsilon, \end{array}$$

com ϵ um número positivo perto de zero.

A metodologia DEA requer a resolução de n programas lineares com q variáveis e $n+1$ restrições, o que pode ser computacionalmente custoso se existirem muitas unidades para analisar. A abordagem DEA pode mostrar-se sensível a mau escalonamento dos dados ou comportar-se menos bem na presença de ruído nos mesmos.

Existem diversas alternativas à abordagem original, da autoria de Charnes, Cooper e Rhodes. Algumas destas alternativas são obtidas modificando, apropriadamente, o dual do programa linear descrito em cima.

Para terminar este brevíssimo resumo sobre a metodologia DEA e para que se possa compreender melhor o seu significado, vamos considerar o caso em que existe um único indicador de *input* para dois de *output*. A fim de simplificar ainda mais as coisas, consideramos *inputs* unitários para todas as unidades. Um caso concreto com cinco unidades seria o seguinte:

Unidade 1	2	4	1
Unidade 2	2	2	1
Unidade 3	4	1	1
Unidade 4	1	2	1
Unidade 5	0.5	2	1

É fácil constatar que as unidades 1 e 3 são pontos estritamente eficientes, se considerarmos o problema de maximizar ambos os *outputs*. Os seus valores de eficiência relativa na abordagem DEA são, ambos, de 100%.

A unidade 2 corresponde a um ponto eficiente (mas não estritamente eficiente) para o mesmo problema bi-objectivo. A sua eficiência relativa na metodologia DEA é de 71.43%. As suas referências de eficiência são as unidades 1 e 3.

As unidades 4 e 5 são dominadas por outras, quando se considera o mesmo problema bi-objectivo. As suas eficiências relativas na metodologia DEA coincidem (50%). Ambas têm por referência única de eficiência a unidade 1.

Aula 7: Modelos de Gestão de Projectos

Os modelos de gestão de projectos constituem um instrumento útil na gestão de projectos relacionados com grandes empreitadas ou operações logísticas complexas. Este tipo de projectos envolve um conjunto de *actividades*, cujo *custo*, *início* ou *duração* importa monitorizar. Os faseamentos das actividades podem estar entre si relacionos. As relações mais frequentes são as de precedência temporal, que se traduzem na exigência de estarem concluídas um subconjunto de actividades antes de ser possível dar início à actividade em consideração.

Exemplo de um Modelo de Gestão de Projectos[1]. Suponhamos que uma empresa de construção civil pretende construir uma obra envolvendo 10 grandes actividades.

Cada actividade tem uma duração mínima e uma duração máxima.

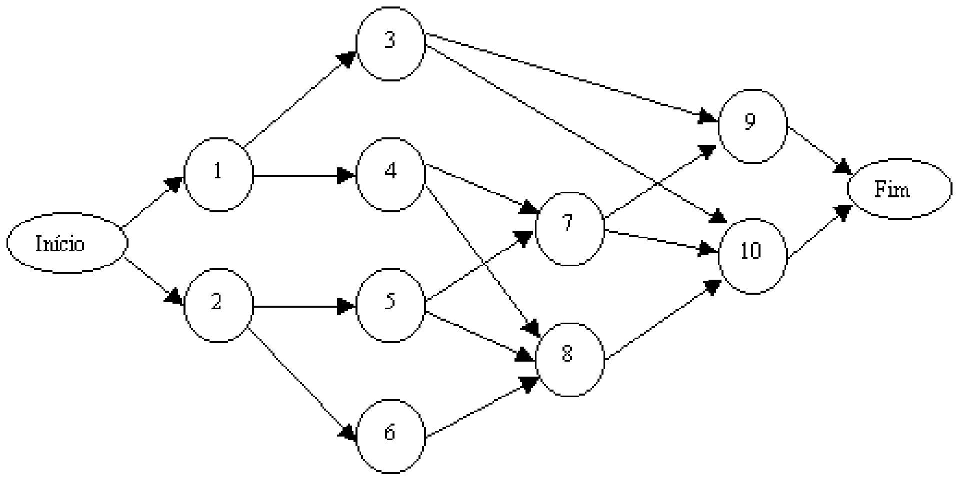
O custo de cada actividade varia linearmente com a sua duração. Conhece-se, por exemplo, o custo associado à duração mínima e o custo atribuído à duração máxima. Este último é o custo regular da actividade. O custo associado à duração mínima é superior ao da duração máxima por necessitar de recursos suplementares (*e.g.*, mão de obra).

Para cada actividade i, lista-se, de seguida, os tempos de duração mínima e máxima (t_i^{min} e t_i^{max}) e os seus respectivos custos (c_i^{min} e c_i^{max}), bem como as actividades que a devem preceder.

Actividade	t_i^{min}	t_i^{max}	c_i^{min}	c_i^{max}	Precedências
1	6	12	8000	5000	—
2	8	16	12000	9000	—
3	16	24	14500	10000	1
4	14	20	9500	6500	1
5	8	12	10000	5500	2
6	6	10	8500	7000	2
7	10	15	15000	10000	4,5
8	4	16	19000	10000	4,5,6
9	12	16	14500	11000	3,7
10	2	12	6500	4000	3,7,8

Os tempos são dados em semanas e os custos em milhares de euros. O grafo das precedências é ilustrado de seguida.

[1] Gentilmente cedido pelo Doutor Pedro Coimbra Martins.



O gestor da obra pretende elaborar um plano para a sua execução que minimize o seu custo total. A duração máxima da obra são 46 semanas.

Seja t_i a variável que mede o tempo gasto na actividade i. Seja d_i a variável que indica a data de início da actividade i. Como as actividades 1 e 2 não dependem de nenhuma outra, podem arrancar imediatamente, ou seja, tem-se que $d_1 = d_2 = 0$.

Dadas as características dos custos especificadas anteriormente, a função objectivo a minimizar é dada por

$$\sum_{i=1}^{10} \left(c_i^{min} - \frac{c_i^{min} - c_i^{max}}{t_i^{max} - t_i^{min}} (t_i - t_i^{min}) \right).$$

Fazendo as contas, obtém-se

$$174750 - 500t_1 - 375t_2 - 562.5t_3 - 500t_4 - 1125t_5 - 375t_6 - 1000t_7 - 750t_8 - 875t_9 - 250t_{10}.$$

As relações de precedência modelam-se, linearmente, requerendo que a data de uma determinada actividade seja não inferior à data da actividade precedente mais o seu tempo de duração. Desta forma, o programa linear que modela esta forma de gerir o projecto é o seguinte:

$$
\begin{aligned}
\text{minimizar} \quad & 174750 - 500t_1 - 375t_2 - 562.5t_3 - 500t_4 - 1125t_5 - 375t_6 - 1000t_7 \\
& -750t_8 - 875t_9 - 250t_{10} \\
\text{sujeito a} \quad & d_1 \;=\; 0, \\
& d_2 \;=\; 0, \\
& d_1 + t_1 \;\le\; d_3, \\
& d_1 + t_1 \;\le\; d_4, \\
& d_2 + t_2 \;\le\; d_5, \\
& d_2 + t_2 \;\le\; d_6, \\
& d_4 + t_4 \;\le\; d_7, \\
& d_5 + t_5 \;\le\; d_7, \\
& d_4 + t_4 \;\le\; d_8, \\
& d_5 + t_5 \;\le\; d_8, \\
& d_6 + t_6 \;\le\; d_8, \\
& d_3 + t_3 \;\le\; d_9, \\
& d_7 + t_7 \;\le\; d_9, \\
& d_3 + t_3 \;\le\; d_{10}, \\
& d_7 + t_7 \;\le\; d_{10}, \\
& d_8 + t_8 \;\le\; d_{10}, \\
& d_9 + t_9 \;\le\; 46, \\
& d_{10} + t_{10} \le 46, \\
& t_i^{min} \;\le\; t_i \;\le\; t_i^{max}, \quad i = 1, \ldots, 10, \\
& d_i \;\ge\; 0, \quad i = 1, \ldots, 10.
\end{aligned}
$$

As variáveis d_1 e d_2 podem ser facilmente eliminadas deste problema.

A solução encontrada foi:

Actividades	1	2	3	4	5	6	7	8	9	10
Inícios	0	0	10	6	8	10	20	20	34	36
Durações	6	8	24	14	12	10	14	16	12	10

O valor óptimo do programa linear é 968750 (custo mínimo da obra em milhares de euros).

Os valores das variáveis de folga correspondentes às restrições de desigualdade são apresentados na tabela seguinte. Cada restrição corresponde a uma relação de precedência, que é explicitada na tabela. Lista-se, também, o preço-sombra associado a cada restrição (ou seja, a componente da solução do dual correspondente à restrição) e os limites inferior e superior do intervalo no qual o coeficiente da restrição pode variar sem alterar a partição em variáveis básicas e não básicas da solução óptima do primal. O preço-sombra actua, somente, entre estes limites.

Precedências	Folgas	Preços–Sombra	Inferior	Superior
$1 \longrightarrow 3$	4	0	-4	$+\infty$
$1 \longrightarrow 4$	0	-500	0	6
$2 \longrightarrow 5$	0	-750	-4	0
$2 \longrightarrow 6$	0	0	-2	$+\infty$
$4 \longrightarrow 7$	2	-500	0	0
$5 \longrightarrow 7$	0	-500	0	0
$4 \longrightarrow 8$	0	0	0	$+\infty$
$5 \longrightarrow 8$	0	-250	-8	0
$6 \longrightarrow 8$	0	0	-2	$+\infty$
$3 \longrightarrow 9$	0	0	-4	2
$7 \longrightarrow 9$	0	-1000	-4	1
$3 \longrightarrow 10$	0	0	-2	$+\infty$
$7 \longrightarrow 10$	2	0	-2	$+\infty$
$8 \longrightarrow 10$	2	-250	-8	2
$9 \longrightarrow$ fim	0	-1000	42	47
$10 \longrightarrow$ fim	0	-250	38	48

A informação associada aos preços-sombra é útil, por exemplo, para a analisar a sensibilidade da obra a variações no prazo dado para completar as actividades terminais (9 e 10). Da penúltima linha desta tabela, conclui-se que a diminuição (resp. aumento) no custo total da obra se o prazo dado para completar a actividade 9 passasse de 46 semanas para 45 semanas (resp. para 47 semanas) seria de 1000 milhares de euros.

Outro tipo de sensibilidade a analisar seria o atraso do arranque de uma actividade. Suponhamos que o início da actividade 9 teria que acontecer quatro semanas depois da actividade 7 terminar. Isso corresponderia a substituir a restrição $d_7 + t_7 \leq d_9$ por $d_7 + t_7 + 4 \leq d_9$, ou seja, por

$$d_7 + t_7 - d_9 \ \leq\ -4.$$

Como o correspondente preço-sombra é -1000, o custo em retardar a actividade 9 em duas semanas seria de 4000 milhares de euros. De forma análoga, o ganho em antecipar a actividade 9 uma semana seria de 1000 milhares de euros. Refira-se que, para esta análise, o 0 do termo independente da restrição $d_7 + t_7 - d_9 \leq 0$ pode variar no intervalo $[-4, 1]$, de acordo com a tabela.

A introdução clássica aos modelos de gestão de projectos costuma incluir as chamadas redes de projecto CPM (*Critical Path Method*). Optámos por uma abordagem, à partida diferente, que nos oferece maior flexibilidade na generalização do modelo e na análise de sensibilidade.

A abordagem CPM trabalha directamente sobre o grafo orientado das precedências, calculando os chamados *caminhos críticos*. O modelo apresentado nesta aula pode ser visto como uma generalização de uma formulação dual associada ao grafo orientado das precedências.

Aula 8: Modelos de Orçamento de Investimentos — Introdução à Programação Linear Inteira

Exemplo de um Modelo de Orçamento de Investimentos. Uma firma pretende investir 10 milhões de euros em imobiliário e está a ponderar a possibilidade de comprar seis imóveis, cujos preços foram negociados. De acordo com um estudo de mercado, o agente responsável pela decisão dispõe de estimativas sobre o preço de mercado desses bens daqui a dez anos. Os preços e as estimativas são dados em baixo, em milhões de euros.

imóveis	1	2	3	4	5	6
preço	4.0	3.8	6.0	7.2	2.0	5.1
estimativa	4.5	4.7	8.0	7.0	4.2	9.2

O modelo pode ser descrito recorrendo a seis variáveis. Cada variável x_j (com $j \in \{1, 2, 3, 4, 5, 6\}$) pode tomar apenas dois valores:

$$x_j = \begin{cases} 1 & \text{se o imóvel } j \text{ for adquirido,} \\ 0 & \text{caso contrário.} \end{cases}$$

A estratégia que maximiza o retorno deste investimento em dez anos é a solução do problema

$$\begin{aligned} \text{maximizar} \quad & 4.5x_1 + 4.7x_2 + 8.0x_3 + 7.0x_4 + 4.2x_5 + 9.2x_6 \\ \text{sujeito a} \quad & 4.0x_1 + 3.8x_2 + 6.0x_3 + 7.2x_4 + 2.0x_5 + 5.1x_6 \leq 10, \\ & x_1, x_2, x_3, x_4, x_5, x_6 \in \{0, 1\}. \end{aligned}$$

Trata-se de um problema de *programação (linear) inteira binária*, conhecido por problema da mochila. As varíaveis podem apenas assumir os valores inteiros 0 ou 1 — binários neste caso. Existe uma única restrição.

A solução óptima deste programa inteiro binário é dada por $x_2 = x_6 = 1$ (com as restantes variáveis iguais a zero). O valor da função objectivo para esta solução é 13.9 (milhões de euros). Se a estimativa do valor de mercado do imóvel 6 descesse para 8.5, a solução óptima passaria a ser $x_1 = x_2 = x_5 = 1$ (com as restantes variáveis nulas), com um valor de 13.4 (milhões de euros).

Neste modelo, a restrição $4.0x_1 + 3.8x_2 + 6.0x_3 + 7.2x_4 + 2.0x_5 + 5.1x_6 \leq 10$ impõe o orçamento disponível para o investimento em causa. Podem existir, porém, mais do que uma restrição orçamental.

Outras alterações ao modelo poderiam incluir *condições de exclusividade*. Suponhamos que os imóveis 2, 4 e 6 estão na posse do mesmo vendedor e que este pretende comercializar apenas um dos seus três bens. Esta situação é modelada pela restrição

$$x_2 + x_4 + x_6 \leq 1.$$

O problema passaria a ser

$$\begin{aligned} \text{maximizar} \quad & 4.5x_1 + 4.7x_2 + 8.0x_3 + 7.0x_4 + 4.2x_5 + 9.2x_6 \\ \text{sujeito a} \quad & 4.0x_1 + 3.8x_2 + 6.0x_3 + 7.2x_4 + 2.0x_5 + 5.1x_6 \ \leq \ 10, \\ & x_2 + x_4 + x_6 \ \leq \ 1, \\ & x_1, x_2, x_3, x_4, x_5, x_6 \in \{0, 1\}, \end{aligned}$$

mantendo a estrutura de um programa inteiro binário, mas deixando de ser um problema da mochila. A solução passaria a ser $x_1 = x_6 = 1$ (com as restantes variáveis nulas), a que corresponde um retorno de 13.7 milhões de euros.

Um programa (linear) inteiro é um problema de optimização da forma

$$\begin{aligned} \text{maximizar} \quad & c^\top x \\ \text{sujeito a} \quad & Ax \ \leq \ b, \\ & x \ \geq \ 0, \\ & x_i \ \in \ \mathbb{Z}, \quad i = 1, \ldots, n, \end{aligned}$$

em que $x \in \mathbb{R}^n$ é o vector das incógnitas, $A \in \mathbb{R}^{m \times n}$ é a matriz das restrições e $b \in \mathbb{R}^m$ é o termo independente das restrições. (No exemplo anterior, $n = 6$ e $m = 1$.) A *relaxação linear* deste programa linear inteiro é o programa linear que se obtém *relaxando* os requisitos de integralidade:

$$\begin{aligned} \text{maximizar} \quad & c^\top x \\ \text{sujeito a} \quad & Ax \ \leq \ b, \\ & x \ \geq \ 0. \end{aligned}$$

A solução da relaxação linear não tem, necessariamente, componentes inteiras. O seu valor óptimo é superior ao valor óptimo do programa linear inteiro. Uma forma de calcular uma solução aproximada para um programa linear inteiro seria arredondar para inteiros as componentes da solução da sua relaxação linear. Colocar-se-ia, de imediato, a questão de como arredondar. Mas mesmo o melhor arredondamento possível (aquele que, de entre todos os admissíveis, correspondeu ao menor valor da função objectivo) pode não ser a solução do programa linear inteiro e, para além disso, pode requerer um cálculo computacionalmente muito dispendioso.

Uma classe de técnicas que determinam uma solução óptima para um programa linear inteiro são os métodos de *branch and bound*, que exploram, se necessário exaustivamente, partes da região admissível onde as variáveis tomam valores inteiros. Note-se que a grande maioria dos problemas de programação linear inteira não apresenta uma complexidade de resolução *polinomial*, ao contrário da dos problemas em programação linear.

Quando as componentes de x apenas puderem tomar os valores 0 ou 1 estamos na presença de um programa (linear) inteiro binário:

$$\begin{aligned} \text{maximizar} \quad & c^\top x \\ \text{sujeito a} \quad & Ax \ \le \ b, \\ & x_i \ \in \ \{0,1\}, \quad i = 1, \ldots, n. \end{aligned}$$

As restrições de não negatividade ($x \ge 0$) são, obviamente, desnecessárias. A relaxação linear deste programa inteiro binário é o programa linear

$$\begin{aligned} \text{maximizar} \quad & c^\top x \\ \text{sujeito a} \quad & Ax \ \le \ b, \\ & x_i \ \in \ [0,1], \quad i = 1, \ldots, n. \end{aligned}$$

Fazer exemplos com duas variáveis no quadro e com o *software*. Explicar o alcançe e a limitação das relaxações lineares.

Exemplo de um Modelo de Orçamento de Projectos. Suponhamos que a administração de um parque industrial necessita de aumentar a área oferecida às empresas instaladas. Para este efeito, considera vários projectos, que podem passar por obras de infra-estrutura ou por um melhor aproveitamento dos terrenos existentes.

Dados os custos envolvidos e a complexidade da operação, a administração sabe que não pode executar todos os projectos de ampliação simultaneamente e considera o seu faseamento em dois anos. Para efectuar as obras, a empresa dispõe de 70 milhares de euros no primeiro ano e de 90 milhares de euros no segundo ano. Os restantes dados do problema são expostos na tabela seguinte (ganhos em área, medidos em m^2, e custos de execução nos dois anos, dados em milhares de euros).

projectos	1	2	3	4	5
área ganha	620	280	100	550	430
custo no ano 1	40	18	2	36	25
custo no ano 2	48	18	15	40	25

Se assumirmos que

$$x_j \ = \ \begin{cases} 1 & \text{se o projecto } j \text{ for escolhido,} \\ 0 & \text{caso contrário,} \end{cases}$$

para $j = 1, 2, 3, 4, 5$, então a estratégia que maximiza o ganho, em área, da intervenção é a solução do problema

$$\begin{aligned} \text{maximizar} \quad & 620x_1 + 280x_2 + 100x_3 + 550x_4 + 430x_5 \\ \text{sujeito a} \quad & 40x_1 + 18x_2 + 2x_3 + 36x_4 + 25x_5 \ \leq \ 70, \\ & 40x_1 + 18x_2 + 15x_3 + 40x_4 + 25x_5 \ \leq \ 90, \\ & x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}. \end{aligned}$$

A solução é dada por $x_1 = x_3 = x_5 = 1$ e $x_2 = x_4 = 0$, com um acréscimo em área de $1150\ m^2$.

É frequente, neste tipo de problemas, existirem *dependências entre projectos*. Por exemplo, para o modelo em causa, a execução do projecto 1 poderia estar dependente, por razões técnicas, da execução do projecto 2. Em tal caso, gostaríamos que se x_1 tomasse o valor 1 então x_2 também fosse igual a 1, exigindo a execução do projecto 2. Esta situação pode ser facilmente modelada através da inclusão, no problema, da restrição

$$x_1 \ \leq \ x_2.$$

O problema de programação inteira binária a resolver passaria a ser

$$\begin{aligned} \text{maximizar} \quad & 620x_1 + 280x_2 + 100x_3 + 550x_4 + 430x_5 \\ \text{sujeito a} \quad & 40x_1 + 18x_2 + 2x_3 + 36x_4 + 25x_5 \ \leq \ 70, \\ & 40x_1 + 18x_2 + 15x_3 + 40x_4 + 25x_5 \ \leq \ 90, \\ & x_1 - x_2 \ \leq \ 0, \\ & x_1, x_2, x_3, x_4, x_5 \in \{0, 1\}. \end{aligned}$$

A solução óptima passaria a ser $x_3 = x_4 = x_5 = 1$ e $x_1 = x_2 = 0$, com um valor da função objectivo de 1080, evitando, deste modo, a execução do projecto 1.

Aula 9: Formas de Modelação em Programação Linear Inteira

Dois dos processos de modelação em programação (linear) inteira foram introduzidos anteriormente, quando foram mencionadas as condições de exclusividade e as relações de dependência, no âmbito da programação inteira binária. Vamos, seguidamente, estudar três outras formas de modelação com recurso a variáveis binárias.

Variáveis com Custo Fixo. Dado um programa linear, escrito numa forma qualquer, consideremos uma alteração à sua formulação em que a contribuição para a função objectivo de uma das variáveis não negativas (a variável x_i), em vez de ser igual a

$$c_i x_i,$$

como até ao momento tem sido estudado, passa a ser dada por

$$\begin{cases} f_i + c_i x_i & \text{se } x_i > 0, \\ 0 & \text{se } x_i = 0. \end{cases}$$

Pretende-se, assim, impor um custo fixo f_i, sempre que $x_i > 0$, uma situação que ocorre com frequência quando x_i está associada ao início de uma actividade ou de um período de funcionamento de uma máquina. Esta contribuição é não linear e descontínua (ao contrário da original que era linear). O novo problema deixa de ser um programa linear.

Seja M um número real positivo *suficientemente grande*. O novo custo associado a x_i pode ser substituído, equivalentemente, por

$$c_i x_i + f_i y_i$$

e

$$x_i \leq M y_i, \quad y_i \in \{0, 1\}.$$

Desta forma, o novo custo passa a ser expresso linearmente na função objectivo. O novo problema fica com mais uma restrição linear ($x_i - M y_i \leq 0$) e com mais uma variável y_i, do tipo binário.

Se, no programa linear original, a variável $x_i \geq 0$ já tivesse um limite superior ($x_i \leq u_i$) então o número M poderia ser substituído por u_i.

No exemplo da Aula 1,

$$\begin{aligned} \text{minimizar} \quad & 24x_1 + 14x_2 + 8x_3 \\ \text{sujeito a} \quad & 4x_1 + 3x_2 + x_3 \geq 100, \\ & 2x_1 + x_2 + x_3 \geq 65, \\ & x_1, x_2, x_3 \geq 0, \end{aligned}$$

suponhamos que o custo de comercializar o equipamento do tipo 3 passaria a ser

$$\begin{cases} 100 + 14x_2 & \text{se } \; x_2 > 0, \\ 0 & \text{se } \; x_2 = 0, \end{cases}$$

com um custo fixo $f_2 = 100$. Neste caso, o programa linear daria lugar ao *programa linear inteiro misto*

$$\begin{array}{rl} \text{minimizar} & 24x_1 + 14x_2 + 8x_3 + 100y_2 \\ \text{sujeito a} & 4x_1 + 3x_2 + x_3 \;\geq\; 100, \\ & 2x_1 + x_2 + x_3 \;\geq\; 65, \\ & x_2 - My_2 \;\leq\; 0, \\ & x_1, x_2, x_3 \geq 0, \quad y_2 \in \{0, 1\}. \end{array}$$

Os valores de x_1, x_2 e x_3 da solução mantêm-se a 0, 17.5 e 47.5, respectivamente. O valor óptimo do programa passa para 725 (correspondendo a $y_2 = 1$). Se o custo fixo f_2 fosse alterado para 200 então as componentes de x na solução passariam a ser 0, 100 e 0 (e o valor óptimo 800, correspondente a $y_2 = 0$).

Um programa linear, em que um ou mais dos custos lineares na função objectivo são substituídos por custos com componente fixa, pode, como acabámos de ver, ser transformado num *programa (linear) inteiro misto*, em que apenas parte das variáveis do problema tomam valores inteiros (neste caso binários). As duas situações seguintes são, igualmente, modeláveis por programação inteira (binária) mista.

Variáveis Tudo-ou-Nada. Os limites inferior e superior de capacidade do tipo $0 \leq x_k \leq u_k$ (estudados nos problemas de fluxo em redes) podem, em determinados casos concretos, dar lugar a uma imposição da forma $\{x_k = 0 \text{ ou } x_k = u_k\}$.

É fácil verificar que a condição $\{x_k = 0 \text{ ou } x_k = u_k\}$ é equivalente a

$$x_k \;=\; u_k y_k, \quad y_k \;\in\; \{0, 1\}.$$

A inclusão da restrição de igualdade $x_k - u_k y_k = 0$ e a introdução da variável binária y_k permitem, desta forma, contornar a natureza *disjuntiva* da imposição.

Restrições Disjuntivas. No enquadramento da programação linear, um problema pode apresentar duas restrições impostas disjuntivamente

$$a_1^\top x \;\leq\; b_1 \quad \text{ou} \quad a_2^\top x \;\leq\; b_2.$$

Se a primeira restrição for satisfeita em x então a segunda não precisa de o ser, e viceversa. Note-se que restrições impostas disjuntivamente,

$$S_d \;=\; \left\{ x \in \mathbb{R}^n : \; a_1^\top x \;\leq\; b_1 \;\text{ ou }\; a_2^\top x \;\leq\; b_2 \right\},$$

relaxam a imposição conjuntiva das mesmas,

$$S_c \;=\; \left\{x \in \mathbb{R}^n :\; a_1^\top x \;\leq\; b_1,\;\; a_2^\top x \;\leq\; b_2\right\},$$

no sentido em que $S_c \subset S_d$.

Seja M um número real positivo suficientemente grande. A obrigatoriedade da satisfação de apenas uma das restrições pode ser expressa, equivalentemente, por

$$a_1^\top x - My_1 \;\leq\; b_1, \quad a_2^\top x - My_2 \;\leq\; b_2, \quad y_1 + y_2 \;=\; 1 \quad \text{e} \quad y_1, y_2 \in \{0,1\}.$$

Este tipo de transformação é aplicável ao caso mais geral em que, de entre m restrições, há somente k restrições (com $k < m$) obrigatoriamente impostas.

De entre os problemas de programação inteira mais utilizados encontram-se os que recorrem às restrições de *cobertura*, de *empacotamento* e de *partição*. Estas restrições aplicam-se a um conjunto de variáveis binárias ($x_j \in \{0,1\}, j = 1, \ldots, n_b$).

As restrições de cobertura (*set covering*) especificam que a solução tem de conter pelo menos um elemento de um subconjunto $J \subset \{1, \ldots, n_b\}$

$$\sum_{j \in J} x_j \;\geq\; 1.$$

As restrições de empacotamento (*set packing*) não permitem que a solução tenha mais do que um elemento de um subconjunto J

$$\sum_{j \in J} x_j \;\leq\; 1.$$

(Refira-se que a condição de exclusividade no modelo de orçamento de investimentos era deste tipo.)

As restrições de partição (*set partitioning*) requerem que a solução tenha um e um só elemento de um subconjunto J

$$\sum_{j \in J} x_j \;=\; 1.$$

Exemplo de Problemas de Cobertura, Partição e Empacotamento. Uma empresa de informação geográfica pretende adquirir equipamento e tem à sua consideração cinco aparelhos com funcionalidades diferentes.

	Aparelhos				
Funcionalidades	1	2	3	4	5
Detecção Remota	×	–	×	–	×
Fotogrametria	–	–	×	×	×
SIG	×	×	×	×	×
Preço (em K euros)	4	5	11	9	15

Se a empresa desejasse minimizar o preço da aquisição, garantindo que todas as funcionalidades fossem cobertas, consideraria a solução do problema de cobertura:

$$\begin{aligned} \text{minimizar} \quad & 4x_1 + 5x_2 + 11x_3 + 9x_4 + 15x_5 \\ \text{sujeito a} \quad & x_1 + x_3 + x_5 \ \geq\ 1, \\ & x_3 + x_4 + x_5 \ \geq\ 1, \\ & x_1 + x_2 + x_3 + x_4 + x_5 \ \geq\ 1, \\ & x_i \in \{0,1\}, \quad i = 1, \ldots, 5. \end{aligned}$$

Se a empresa pretendesse minimizar o preço da aquisição, exigindo que as funcionalidades fossem satisfeitas por um e um só dos aparelhos, resolveria o problema de partição:

$$\begin{aligned} \text{minimizar} \quad & 4x_1 + 5x_2 + 11x_3 + 9x_4 + 15x_5 \\ \text{sujeito a} \quad & x_1 + x_3 + x_5 \ =\ 1, \\ & x_3 + x_4 + x_5 \ =\ 1, \\ & x_1 + x_2 + x_3 + x_4 + x_5 \ =\ 1, \\ & x_i \in \{0,1\}, \quad i = 1, \ldots, 5. \end{aligned}$$

(O valor óptimo deste programa linear inteiro não pode ser inferior ao do anterior, pois ao passar uma igualdade a desigualdade estamos a *relaxar* a região admissível.)

Se os números dados para os preços fossem indicadores de qualidade e a empresa estivesse interessada na compra que melhor qualidade oferecesse, com, no máximo, um aparelho para cada funcionalidade, então formularia o seguinte problema de empacotamento:

$$\begin{aligned} \text{maximizar} \quad & 4x_1 + 5x_2 + 11x_3 + 9x_4 + 15x_5 \\ \text{sujeito a} \quad & x_1 + x_3 + x_5 \ \leq\ 1, \\ & x_3 + x_4 + x_5 \ \leq\ 1, \\ & x_1 + x_2 + x_3 + x_4 + x_5 \ \leq\ 1, \\ & x_i \in \{0,1\}, \quad i = 1, \ldots, 5. \end{aligned}$$

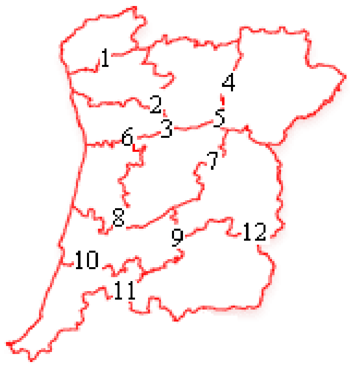
Exemplo de um Problema de Cobertura em Planeamento de Localizações. A administração central pretente equipar os distritos das regiões norte e centro com unidades geradoras de energia eólica, considerando a colocação de 12 geradores. Cada gerador pode servir mais do que um distrito, de acordo com a figura. O número mínimo de geradores necessários é o valor óptimo do problema de cobertura:

$$
\begin{array}{rll}
\text{minimizar} & \sum_{i=1}^{12} x_i & \\
\text{sujeito a} & x_1 \ \geq \ 1, & \text{Viana do Castelo,} \\
& x_1 + x_2 \ \geq \ 1, & \text{Braga,} \\
& x_2 + x_3 + x_6 \ \geq \ 1, & \text{Porto,} \\
& x_2 + x_3 + x_4 + x_5 \ \geq \ 1, & \text{Vila Real,} \\
& x_4 + x_5 \ \geq \ 1, & \text{Bragança,} \\
& x_6 + x_8 \ \geq \ 1, & \text{Aveiro,} \\
& x_3 + x_5 + x_6 + x_7 + x_8 \ \geq \ 1, & \text{Viseu,} \\
& x_5 + x_7 + x_9 + x_{12} \ \geq \ 1, & \text{Guarda,} \\
& x_8 + x_9 + x_{10} \ \geq \ 1, & \text{Coimbra,} \\
& x_{10} + x_{11} \ \geq \ 1, & \text{Leiria,} \\
& x_9 + x_{11} + x_{12} \ \geq \ 1, & \text{Castelo Branco,} \\
& x_i \in \{0,1\}, \quad i = 1, \ldots, 12. &
\end{array}
$$

A solução deste programa inteiro binário é dada por:

$$x_1 = x_4 = x_6 = x_{10} = x_{12} = 1 \quad \text{e} \quad x_2 = x_3 = x_5 = x_7 = x_8 = x_9 = x_{11} = 0.$$

Os geradores 1, 4, 6, 10 e 12 seriam os escolhidos. O valor óptimo do problema é igual a 5.


1
2
3
4
5
6
7
8
9
10
11
12

Aula 10: Problemas de Afectação

Quando estudámos os problemas de fluxos em redes, omitimos um dos problemas mais importantes desta classe, o *problema da afectação.*

Exemplo de um Problema de Afectação (1). Consideremos a elaboração de um plano de trabalho, em que se pretende afectar 5 trabalhadores a 5 tarefas, de forma a que cada tarefa seja feita por um e um só trabalhador e que cada trabalhador faça uma e uma só tarefa. Na instância do problema em causa, o objectivo é minimizar o custo total da afectação (medido em centenas de euros). Um objectivo alternativo seria maximizar a produtividade total.

Os custos em afectar os trabalhadores às tarefas são dados na tabela em baixo. Um custo correspondente a um par tarefa–trabalhador é omitido quando não é possível (ou admissível) afectar esse trabalhador a essa tarefa .

	Trabalhadores				
Tarefas	1	2	3	4	5
1	15	10	–	–	12
2	7	15	18	9	5
3	9	–	15	9	7
4	–	7	–	10	3
5	8	–	–	11	–

Este problema é modelado através do programa inteiro binário:

$$\begin{array}{rl} \text{minimizar} & 15x_{11} + 10x_{12} + 12x_{15} + 7x_{21} + 15x_{22} + 18x_{23} + 9x_{24} + 5x_{25} + 9x_{31} \\ & + 15x_{33} + 9x_{34} + 7x_{35} + 7x_{42} + 10x_{44} + 3x_{45} + 8x_{51} + 11x_{54} \\ \text{sujeito a} & x_{11} + x_{12} + x_{15} = 1, \\ & x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 1, \\ & x_{31} + x_{33} + x_{34} + x_{35} = 1, \\ & x_{42} + x_{44} + x_{45} = 1, \\ & x_{51} + x_{54} = 1, \\ & x_{11} + x_{21} + x_{31} + x_{51} = 1, \\ & x_{12} + x_{22} + x_{42} = 1, \\ & x_{23} + x_{33} = 1, \\ & x_{24} + x_{34} + x_{44} + x_{54} = 1, \\ & x_{15} + x_{25} + x_{35} + x_{45} = 1, \\ & x_{ij} \in \{0,1\} \quad \text{para } (i,j) \text{ admissível.} \end{array}$$

As cinco primeiras restrições asseguram que cada tarefa é feita por um e um só trabalhador. As cinco restrições seguintes garantem que cada trabalhador faz uma e uma só tarefa. As variáveis do problema tomam valores binários:

$$x_{ij} = \begin{cases} 1 & \text{se a tarefa } i \text{ é feita pelo trabalhador } j, \\ 0 & \text{caso contrário,} \end{cases}$$

para todo o par ordenado (i, j) correspondente a uma atribuição admissível.

A solução encontrada tem o valor objectivo 45 e consiste em tomar:

$$x_{12} = x_{24} = x_{33} = x_{45} = x_{51} = 1, \quad \text{com as restantes iguais a zero.}$$

De uma forma geral, o problema da afectação pode ser escrito como

$$\begin{array}{rl} \text{minimizar} & \sum_{(i,j) \text{ admissível}} c_{ij} x_{ij} \\ \text{sujeito a} & \sum_{j:(i,j) \text{ admissível}} x_{ij} \ = \ 1, \quad i \in I, \\ & \sum_{i:(i,j) \text{ admissível}} x_{ij} \ = \ 1, \quad j \in J, \\ & x_{ij} \ \in \ \{0,1\} \quad \text{para } (i,j) \text{ admissível,} \end{array}$$

em que os índices i e j pertencem a dois conjuntos finitos I e J com o mesmo cardinal.

Quando não existem pares inadmissíveis, o problema de afectação reduz-se à sua *forma padrão*

$$\begin{array}{rl} \text{minimizar} & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\ \text{sujeito a} & \sum_{j \in J} x_{ij} \ = \ 1, \quad i \in I, \\ & \sum_{i \in I} x_{ij} \ = \ 1, \quad j \in J, \\ & x_{ij} \ \in \ \{0,1\}, \quad i \in I, \quad j \in J. \end{array}$$

A relaxação linear deste programa inteiro binário é dada por:

$$\begin{array}{rl} \text{minimizar} & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\ \text{sujeito a} & \sum_{j \in J} x_{ij} \ = \ 1, \quad i \in I, \\ & \sum_{i \in I} x_{ij} \ = \ 1, \quad j \in J, \\ & 0 \ \leq \ x_{ij} \ \leq \ 1, \quad i \in I, \quad j \in J. \end{array}$$

A matriz das restrições deste programa linear, por ser uma matriz de incidência nodo-arco, goza da propriedade de unimodularidade total. Assim, e porque o termo independente tem componentes inteiras, este programa linear tem solução óptima com componentes inteiras. Dito por outras palavras, a solução da relaxação linear do programa de afectação é uma solução do próprio problema de afectação (uma ocorrência afortunada em programação linear inteira...).

Além disso, a relaxação linear do problema de afectação pode ser simplificada, uma vez que as restrições de igualdade impedem que as variáveis tomem valores maiores que 1. Deste modo, a relaxação linear é equivalente ao programa linear:

$$\begin{array}{rrcl}
\text{minimizar} & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} & & \\
\text{sujeito a} & \sum_{j\in J} x_{ij} & = & 1, \quad i\in I, \\
& \sum_{i\in I} x_{ij} & = & 1, \quad j\in J, \\
& x_{ij} & \geq & 0, \quad i\in I, \quad j\in J.
\end{array}$$

Observamos que o problema da afectação (na sua forma padrão) é um caso particular do problema de transportes com provisões e procuras unitários (e, assim sendo, constitui um problema de fluxo de custo mínimo).

Exemplo de um Problema de Afectação (2). Regressemos ao exemplo anterior e suponhamos que existiam mais três trabalhadores do que tarefas. A nova tabela de custos passaria a ter mais três colunas:

	Trabalhadores							
Tarefas	1	2	3	4	5	6	7	8
1	15	10	–	–	12	20	–	13
2	7	15	18	9	5	–	10	9
3	9	–	15	9	7	8	12	10
4	–	7	–	10	3	9	–	15
5	8	–	–	11	–	16	10	12

Este problema é modelado através do programa inteiro binário:

$$\begin{array}{rl}
\text{minimizar} & \textit{função objectivo original} + 20x_{16} + 13x_{18} + 10x_{27} + 9x_{28} \\
& + 8x_{36} + 12x_{37} + 10x_{38} + 9x_{46} + 5x_{48} + 16x_{56} + 10x_{57} + 12x_{58} \\
\text{sujeito a} & x_{11} + x_{12} + x_{15} + (x_{16} + x_{18}) \ = \ 1, \\
& x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + (x_{27} + x_{28}) \ = \ 1, \\
& x_{31} + x_{33} + x_{34} + x_{35} + (x_{36} + x_{37} + x_{38}) \ = \ 1, \\
& x_{42} + x_{44} + x_{45} + (x_{46} + x_{48}) \ = \ 1, \\
& x_{51} + x_{54} + (x_{56} + x_{57} + x_{58}) \ = \ 1, \\
& \textit{(restrições originais relativas aos primeiros 5 trabalhadores)}, \\
& x_{16} + x_{36} + x_{46} + x_{56} \ = \ 1, \\
& x_{27} + x_{37} + x_{57} \ = \ 1, \\
& x_{18} + x_{28} + x_{38} + x_{48} + x_{58} \ = \ 1, \\
& x_{ij} \in \{0,1\} \quad \text{para } (i,j) \text{ admissível.}
\end{array}$$

Este problema tem uma região admissível vazia, como acontece nos problemas de transportes, quando a soma das quantidades procuradas não coincide com a soma das quantidades a fornecer. Tal como sucede nos problemas de fluxo de custo mínimo em redes, uma forma de contornar a situação passaria por acrescentar três tarefas (nodos) artificiais 6, 7 e 8 com ligações admissíveis a todos os trabalhadores e com custos nulos na função objectivo:

$$
\begin{array}{rl}
\text{minimizar} & \textit{função objectivo anterior} \\
\text{sujeito a} & \textit{(restrições anteriores relativas às 5 tarefas)}, \\
& x_{61}+x_{62}+x_{63}+x_{64}+x_{65}+x_{66}+x_{67}+x_{68} \;=\; 1 \quad \textit{(tarefa artificial)}, \\
& x_{71}+x_{72}+x_{73}+x_{74}+x_{75}+x_{76}+x_{77}+x_{78} \;=\; 1 \quad \textit{(tarefa artificial)}, \\
& x_{81}+x_{82}+x_{83}+x_{84}+x_{85}+x_{86}+x_{87}+x_{88} \;=\; 1 \quad \textit{(tarefa artificial)}, \\
& x_{11}+x_{21}+x_{31}+x_{51}+(x_{61}+x_{71}+x_{81}) \;=\; 1, \\
& x_{12}+x_{22}+x_{42}+(x_{62}+x_{72}+x_{82}) \;=\; 1, \\
& x_{23}+x_{33}+(x_{63}+x_{73}+x_{83}) \;=\; 1, \\
& x_{24}+x_{34}+x_{44}+x_{54}+(x_{64}+x_{74}+x_{84}) \;=\; 1, \\
& x_{15}+x_{25}+x_{35}+x_{45}+x_{55}+(x_{65}+x_{75}+x_{85}) \;=\; 1, \\
& x_{16}+x_{36}+x_{46}+x_{56}+(x_{66}+x_{76}+x_{86}) \;=\; 1, \\
& x_{27}+x_{37}+x_{57}+(x_{67}+x_{77}+x_{87}) \;=\; 1, \\
& x_{18}+x_{28}+x_{38}+x_{48}+x_{58}+(x_{68}+x_{78}+x_{88}) \;=\; 1, \\
& x_{ij} \in \{0,1\} \quad \text{para } (i,j) \text{ admissível.}
\end{array}
$$

A solução encontrada tem o valor objectivo 36 e consiste em fazer:

$$x_{12} = x_{25} = x_{36} = x_{48} = x_{51} = 1, \quad \text{com as restantes iguais a zero.}$$

Terminamos este resumo com os chamados *problemas de afectação generalizados*, que ilustraremos recorrendo ao problema de afectação com 5 tarefas e 8 trabalhadores dado em cima. Nesta nova variante, cada tarefa pode ficar por fazer ou ser feita por mais do que um trabalhador. Desta forma, as restrições associadas às tarefas são retiradas do problema. Deixam de ser necessárias as três tarefas artificiais. No entanto, cada trabalhador continua a realizar uma e uma só tarefa (e, assim, as restrições correspondentes aos trabalhadores permanecem no problema).

Estes problemas de afectação generalizada incluem restrições de capacidade especiais. No nosso problema, por exemplo, as capacidades dizem respeito a restrições de tempo impostas às tarefas. Vamos supor que as tarefas 2 e 5 têm um limite de tempo de 12 horas. Os tempos, em horas, que cada trabalhador demora a efectuar estas duas tarefas são dados em baixo.

Trabalhadores	1	2	3	4	5	6	7	8
Tarefa 2	10	20	15	12	9	–	20	18
Tarefa 5	20	–	–	10	–	5	13	15

O novo programa inteiro binário passa a ser o seguinte:

$$\begin{aligned}
\text{minimizar} \quad & \textit{função objectivo anterior} \\
\text{sujeito a} \quad & \textit{(restrições originais relativas aos primeiros 5 trabalhadores)}, \\
& x_{16} + x_{36} + x_{46} + x_{56} \ = \ 1, \\
& x_{27} + x_{37} + x_{57} \ = \ 1, \\
& x_{18} + x_{28} + x_{38} + x_{48} + x_{58} \ = \ 1, \\
& 10x_{21} + 20x_{22} + 15x_{23} + 12x_{24} + 9x_{25} + 20x_{27} + 18x_{28} \ \leq \ 12, \\
& 20x_{51} + 10x_{54} + 5x_{56} + 13x_{57} + 15x_{58} \ \leq \ 12, \\
& x_{ij} \in \{0, 1\} \quad \text{para } (i, j) \text{ admissível.}
\end{aligned}$$

A solução encontrada tem o valor objectivo 66 e consiste em tomar:

$$x_{21} = x_{42} = x_{33} = x_{34} = x_{45} = x_{36} = x_{37} = x_{48} = 1, \quad \text{com as restantes iguais a zero.}$$

Relaxando as duas restrições de capacidade, o valor óptimo desce para 64, correspondendo a

$$x_{21} = x_{42} = x_{33} = x_{24} = x_{45} = x_{36} = x_{27} = x_{48} = 1, \quad \text{com as restantes iguais a zero.}$$

Os problemas de afectação generalizada são menos tratáveis computacionalmente do que os problemas de afectação. Em particular, uma solução da relaxação linear de um problema de afectação generalizado pode não ter componentes inteiras.

Aula 11: Modelos de Localização de Equipamentos e de Projecto de Redes

A programação linear inteira (ou inteira mista) permite modelar inúmeras situações relacionadas com a gestão de equipamentos ou instalações e a gestão de redes de comunicação ou distribuição.

Os instrumentos de modelação em programação linear inteira vão ser aplicados, nesta aula, a dois tipos de problemas com vasta aplicação (a localização de equipamentos e o projecto de redes de distribuição).

Existem situações mais complexas, onde se pretende, simultaneamente, *localizar* centros de distribuição e *projectar* as respectivas redes de distribuição. Os dois modelos apresentados poderiam ser adaptados e combinados num modelo único para dar resposta a estas situações.

Exemplo de um Modelo de Localização de Equipamentos. Consideremos um distrito com 18 concelhos nos quais se pretende instalar 6 centrais de distribuição de medicamentos.

São conhecidos os custos fixos de funcionamento de cada central. Se a central i for aberta então incorre num custo fixo de funcionamento f_i e, nesse caso, pode enviar até u_i unidades de medicamentos.

Fazem parte dos dados do problema, os custos de transporte unitários das centrais para os centros de saúde em cada concelho (c_{ij}), bem como as necessidades de medicamento de cada centro de sáude (d_j).

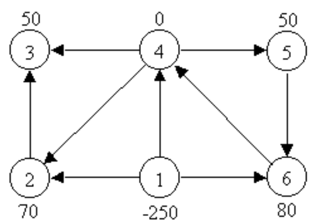
O objectivo do problema é determinar quais os centros que deverão ser abertos de forma a minimizar o custo total da operação. São necessárias variáveis x_{ij}, contínuas e não negativas, para descrever a quantidade a enviar da central i para o concelho j e variáveis binárias y_i para indicar se uma central i está aberta ou fechada. O programa linear inteiro misto que modela este problema pode ser escrito na forma:

$$
\begin{array}{rl}
\text{minimizar} & \sum_{i=1}^{6} \sum_{j=1}^{18} c_{ij} x_{ij} + \sum_{i=1}^{6} f_i y_i \\
\text{sujeito a} & \sum_{i=1}^{6} x_{ij} \;=\; d_j, \quad j = 1, \ldots, 18, \\
& \sum_{j=1}^{18} x_{ij} \;\leq\; u_i y_i, \quad i = 1, \ldots, 6, \\
& x_{ij} \;\geq\; 0, \quad i = 1, \ldots, 6, \quad j = 1, \ldots, 18, \\
& y_i \;\in\; \{0, 1\}, \quad i = 1, \ldots, 6.
\end{array}
$$

Este problema poderia admitir uma versão puramente binária se os medicamentos fossem empacotados em caixotes ou contentores e as necessidades dos centros de saúde fossem expressas em número de contentores, por exemplo, $d_j = 1$, $j = 1, \ldots, 18$. Neste exemplo, as variáveis x_{ij} seriam restringidas aos valores 0 e 1.

A seguir, vamos apresentar um modelo simplificado de um projecto de redes (tradução do inglês *network design*).

Exemplo de um Modelo de Projecto de Redes. Uma empresa de telecomunicações estuda a possibilidade distribuir dados de uma cidade (1) para quatro outras cidades (2, 3, 5 e 6) através da instalação de uma rede de cabo, de acordo com a figura em baixo. O modelo inclui um posto de redistribuição (4), que funciona como um nodo de entreposto.



As ligações entre as cidades podem não ser estabelecidas. Uma ligação entre as cidades i e j acarreta um custo fixo (f_{ij}). Por cada unidade de dados que for transmitida de i para j, incorre-se num um custo unitário de c_{ij}. As ligações têm uma largura de banda máxima (u_{ij}). Os dados deste problema são descritos na tabela:

Ligação	(1,2)	(1,4)	(1,6)	(2,3)	(4,2)	(4,3)	(4,5)	(5,6)	(6,4)
Custos Variáveis (c_{ij})	20	20	50	20	10	10	10	20	20
Custos Fixos (f_{ij})	800	800	1000	800	500	500	500	800	800
Capacidades (u_{ij})	150	200	150	150	200	200	200	200	150

Para formular o problema, associam-se duas variáveis a cada ligação ou arco. A primeira variável ($x_{ij} \geq 0$) descreve a quantidade de dados a enviar de i para j quando a segunda variável ($y_{ij} \in \{0, 1\}$) tomar o valor 1. Se $y_{ij} = 0$ a ligação $i \longrightarrow j$ não é estabelecida.

Se a empresa desejar a solução de custo mínimo deverá formular o programa linear inteiro misto:

$$
\begin{aligned}
\text{minimizar} \quad & \textstyle\sum_{(i,j)\ \text{admissível}} [c_{ij}x_{ij} + f_{ij}y_{ij}] \\
\text{sujeito a} \quad & -x_{12} - x_{14} - x_{16} \ = \ -250, \\
& x_{12} + x_{42} - x_{23} \ = \ 70, \\
& x_{23} + x_{43} \ = \ 50, \\
& x_{14} + x_{64} - x_{42} - x_{43} - x_{45} \ = \ 0, \\
& x_{45} - x_{56} \ = \ 50, \\
& x_{16} + x_{56} - x_{64} \ = \ 80, \\
& 0 \ \le \ x_{ij} \ \le \ u_{ij}y_{ij} \quad \text{para } (i,j) \text{ admissível}, \\
& y_{ij} \ \in \ \{0,1\} \quad \text{para } (i,j) \text{ admissível}.
\end{aligned}
$$

As seis primeiras restrições correspondem ao fluxo de dados a distribuir, enquanto que as últimas nove impõem as restrições associadas às larguras de banda das ligações.

A solução obtida foi:

(i,j)	(1,2)	(1,4)	(1,6)	(2,3)	(4,2)	(4,3)	(4,5)	(5,6)	(6,4)
y_{ij}	1	1	0	0	0	1	1	1	0
x_{ij}	70	180	0	0	0	50	130	80	0

O valor óptimo é de 11800. Se o custo fixo associado à ligação $(1,6)$ descesse de 1000 para 800 passaria a existir uma solução alternativa (com o mesmo valor óptimo), dada por:

(i,j)	(1,2)	(1,4)	(1,6)	(2,3)	(4,2)	(4,3)	(4,5)	(5,6)	(6,4)
y_{ij}	1	1	1	0	0	1	1	0	0
x_{ij}	70	100	80	0	0	50	50	0	0

Aula 12: Optimização de Rotas de Veículos

A determinação de percursos óptimos em transporte de pessoas e de mercadorias constitui um aspecto relevante numa política integrada de redução de custos em transportes. Nas autarquias ou nas áreas metropolitanas, por exemplo, existe a necessidade de optimizar os recursos disponíveis associados aos transportes públicos rodoviários e ao transporte de resíduos sólidos urbanos, reduzindo os custos de combustível, laborais ou outros.

Os modelos de optimização de rotas de veículos são, de uma forma geral, complexos, envolvendo um elevado número de variáveis inteiras ou binárias. As suas formulações assentam em componentes associadas a vários submodelos particulares. Um dos submodelos que aparece com mais frequência está relacionado com o problema do caixeiro viajante, em inglês *Traveling Salesman Problem* (TSP), ou suas variações.

No *problema do caixeiro viajante*, é dado um conjunto de n nodos e são especificadas as distâncias entre eles. O objectivo é calcular um percurso de distância mínima que passe por cada nodo uma única vez.

Vamos apresentar, apenas, a versão simétrica do TSP, em que a distância do nodo i ao nodo j é igual à distância entre o nodo j e o nodo i ($d_{ij} = d_{ji}$), para $i = 1, \ldots, n$ e $j = 1, \ldots, n$ ($i \neq j$).

Dada a simetria das distâncias entre nodos, as variáveis do problema,

$$x_{ij} = \begin{cases} 1 & \text{quando um percurso inclui a ligação entre } i \text{ e } j, \\ 0 & \text{caso contrário,} \end{cases}$$

são definidas apenas para os índices i e j tais que $i < j$. O TSP simétrico apresenta a formulação:

$$\begin{array}{rl} \text{minimizar} & \sum_{i=1}^{n} \sum_{j=1,j>i}^{n} d_{ij}x_{ij} \\ \text{sujeito a} & \sum_{j=1,j<i}^{n} x_{ji} + \sum_{j=1,j>i}^{n} x_{ij} \;=\; 2, \quad i = 1, \ldots, n, \\ & \text{restrições impeditivas de subpercursos,} \\ & x_{ij} \;\in\; \{0, 1\}, \quad j > i. \end{array}$$

As primeiras n restrições asseguram que quando existe um j tal que $x_{ji} = 1$ então também existe um ℓ tal que $x_{i\ell} = 1$. Estas restrições fazem com que se chegue sempre ao mesmo nodo de onde se partiu, mas não evitam a formação de subpercursos com menos de n nodos. O segundo grupo de restrições impede a formação de subpercursos que, satisfazendo as primeiras n restrições, não formem um único percurso envolvendo *todos* os n nodos. O número destas restrições varia factorialmente com n.

Exemplo de um TSP simétrico. Quando $n = 6$ o TSP simétrico, formula-se na forma:

$$\begin{array}{rl} \text{minimizar} & d_{12}x_{12} + d_{13}x_{13} + d_{14}x_{14} + d_{15}x_{15} + d_{16}x_{16} \\ & + \, d_{23}x_{23} + d_{24}x_{24} + d_{25}x_{25} + d_{26}x_{26} \\ & + \, d_{34}x_{34} + d_{35}x_{35} + d_{36}x_{36} + d_{45}x_{45} + d_{46}x_{46} + d_{56}x_{56} \end{array}$$

$$
\begin{aligned}
\text{sujeito a} \quad & x_{12} + x_{13} + x_{14} + x_{15} + x_{16} \;=\; 2, \\
& x_{12} + x_{23} + x_{24} + x_{25} + x_{26} \;=\; 2, \\
& x_{13} + x_{23} + x_{34} + x_{35} + x_{36} \;=\; 2, \\
& x_{14} + x_{24} + x_{34} + x_{45} + x_{46} \;=\; 2, \\
& x_{15} + x_{25} + x_{35} + x_{45} + x_{56} \;=\; 2, \\
& x_{16} + x_{26} + x_{36} + x_{46} + x_{56} \;=\; 2, \\
& \text{restrições impeditivas de subpercursos}, \\
& x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{23}, x_{24}, x_{25}, x_{26}, x_{34}, x_{35}, x_{36}, \\
& x_{45}, x_{46}, x_{56} \;\in\; \{0,1\}.
\end{aligned}
$$

As restrições impeditivas de subpercursos teriam que actuar sobre todos os subpercursos com três nodos (os únicos possíveis a satisfazer as primeiras seis restrições, uma vez que $n = 6$). Consideremos os dois subpercursos correspondentes aos subconjuntos de nodos $\{1,2,3\}$ e $\{4,5,6\}$. As variáveis tomariam, neste caso, os valores:

$$x_{12} = x_{23} = x_{13} = 1 \quad \text{e} \quad x_{45} = x_{46} = x_{45} = 1$$

(com as restantes a valer zero). Estes valores das variáveis satisfazem as primeiras seis restrições do problema. A restrição impeditiva de subpercursos que estes valores não satisfariam é a seguinte:

$$(x_{14} + x_{24} + x_{34}) + (x_{15} + x_{25} + x_{35}) + (x_{16} + x_{26} + x_{36}) \;\geq\; 2.$$

Basicamente, esta restrição obriga a que o que entra e sai de $\{1,2,3\}$ não possa ser inferior a 2. Este problema apresentaria

$$\frac{1}{2}\begin{pmatrix} 6 \\ 3 \end{pmatrix} \;=\; 10$$

restrições impeditivas de subpercursos.

Os modelos de optimização de rotas envolvem, usualmente, um conjunto de veículos, cujos percursos importa determinar de forma óptima. Os modelos conhecidos diferem entre si em variadíssimos aspectos. Em muitos casos, o número de veículos é um dado do problema. Noutros, o número de veículos é uma incógnita do problema. É suposto os veículos passarem por um conjunto de nodos para entrega ou recolha, sendo variadas as restrições de tempo ou capacidade impostas aos veículos. Os próprios pontos de partida e de chegada dos veículos podem ser iguais ou diferentes (ou coincidirem com pontos de recolha).

O exemplo seguinte descreve um modelo prático para a determinação de percursos óptimos em recolha de resíduos sólidos urbanos.

Exemplo de um Modelo de Optimização de Rotas de Veículos. Os dados do problema são os seguintes:

- um ponto ou nodo (0) de partida e de chegada de todos os veículos;
- n pontos ou nodos de recolha $(1, \ldots, n)$, com quantidades a recolher dadas por r_i, $i = 1, \ldots, n$, e tempos de recolha definidos por t_i, $i = 1, \ldots, n$;
- m veículos de recolha $(1, \ldots, m)$ com capacidades dadas por U_k, $k = 1, \ldots, m$, e tempos máximos de recolha dados por T_k, $k = 1, \ldots, m$;
- as distâncias ou custos entre todos os nodos $(c_{ij},\ i = 0, \ldots, n,\ j = 0, \ldots, n)$.

O custo c_{ij} pode ser, por exemplo, a distância mais curta entre o nodo i e o nodo j. Este custo é, geralmente, simétrico, no sentido em que $c_{ij} = c_{ji}$ (o que é o caso se as distâncias entre todos os nodos forem, igualmente, simétricas).

Existem várias formulações para este problema. Adoptamos uma formulação que se baseia nas variáveis binárias

$$y_i^k \;=\; \begin{cases} 1 & \text{se o veículo } k \text{ recolheu no ponto } i, \\ 0 & \text{caso contrário,} \end{cases}$$

$i = 0, \ldots, n$, $k = 1, \ldots, m$, e

$$x_{ij}^k \;=\; \begin{cases} 1 & \text{se o veículo } k \text{ faz a ligação } i \longrightarrow j \text{, recolhendo em ambos,} \\ 0 & \text{caso contrário,} \end{cases}$$

$i = 0, \ldots, n$, $j = 0, \ldots, n$, $k = 1, \ldots, m$. Dada a simetria dos custos, apenas introduziremos estas variáveis para índices i e j tais que $i < j$.

Com as variáveis y_i^k, é fácil formular as restrições de capacidade

$$\sum_{i=1}^{n} r_i\, y_i^k \;\leq\; U_k, \quad k = 1, \ldots, m,$$

e de tempo

$$\sum_{i=1}^{n} t_i\, y_i^k \;\leq\; T_k, \quad k = 1, \ldots, m,$$

de todos os veículos. É preciso impor, também, que cada ponto seja recolhido por um e um só veículo

$$\sum_{k=1}^{m} y_i^k \;=\; 1, \quad i = 1, \ldots, n$$

e que todos os veículos partam do ponto de partida e cheguem a este mesmo

$$\sum_{k=1}^{m} y_0^k \;=\; m.$$

Para calcular o custo total de toda a recolha é preciso identificar o custo de recolha de cada veículo. Este custo, por sua vez, está associado ao percurso que o veículo faz, partindo do nodo 0 e chegando ao nodo 0, para visitar todos os pontos em que recolhe. Deste modo, surgem as restrições associadas a um TSP para cada veículo k, $k = 1, \ldots, m$,

$$\sum_{j=0, j<i}^{n} x_{ji}^k + \sum_{j=0, j>i}^{n} x_{ij}^k \; = \; 2y_i^k, \quad i = 0, \ldots, n.$$

Refira-se que quando $y_i^k = 0$, ou seja, que quando o veículo k não recolhe em i, a restrição correspondente obriga a que todas as variáveis envolvidas sejam nulas. O modelo teria que incluir ainda as restrições impeditivas de subpercursos.

Finalmente, a função objectivo do programa inteiro binário mede o custo de todos os percursos de recolha

$$\sum_{i=0}^{n} \sum_{j=0, j>i}^{n} c_{ij} \left(\sum_{k=1}^{m} x_{ij}^k \right).$$

Reunindo todas as restrições, formulamos o problema da seguinte forma:

$$\begin{array}{rl}
\text{minimizar} & \sum_{i=0}^{n} \sum_{j=0,j>i}^{n} c_{ij} \left(\sum_{k=1}^{m} x_{ij}^k \right) \\
\text{sujeito a} & \sum_{k=1}^{m} y_i^k \; = \; 1, \quad i = 1, \ldots, n, \\
& \sum_{k=1}^{m} y_0^k \; = \; m, \\
& \sum_{i=1}^{n} r_i \, y_i^k \; \leq \; U_k, \quad k = 1, \ldots, m, \\
& \sum_{i=1}^{n} t_i \, y_i^k \; \leq \; T_k, \quad k = 1, \ldots, m, \\
& \sum_{j=0,j<i}^{n} x_{ji}^k + \sum_{j=0,j>i}^{n} x_{ij}^k \; = \; 2y_i^k, \quad i = 0, \ldots, n, \quad k = 1, \ldots, m, \\
& \text{restrições impeditivas de subpercursos}, \\
& y_i^k \; \in \; \{0, 1\}, \quad i = 0, \ldots, n, \quad k = 1, \ldots, m, \\
& x_{ij}^k \; \in \; \{0, 1\}, \quad i = 0, \ldots, n, \quad j = 0, \ldots, n, \quad j > i, \quad k = 1, \ldots, m.
\end{array}$$

Bibliografia

1. *Casos Práticos da Investigação Operacional*, coordenação de C. Henggeler Antunes e L. Valadares Tavares, McGraw-Hill, Lisboa, 2000.

2. V. Chvátal, *Linear Programming*, W. H. Freeman, Nova Iorque, 1983.

3. F. S. Hillier e G. J. Lieberman, *Introduction to Operations Research*, sétima edição, McGraw-Hill, Boston, 2001.

4. S. G. Nash e A. Sofer, *Linear and Nonlinear Programming*, McGraw-Hill, Nova Iorque, 1996.

5. M. Ramalhete, J. Guerreiro e A. Magalhães, *Programação Linear*, Vol. I, McGraw-Hill, Lisboa, 1984. J. Guerreiro, A. Magalhães e M. Ramalhete, *Programação Linear*, Vol. II, McGraw-Hill, Lisboa, 1985.

6. R. L. Rardin, *Optimization in Operations Research*, Prentice Hall, Upper Saddle River, 1998.

7. H. P. Williams, *Model Building in Mathematical Programming*, terceira edição, John Wiley & Sons, Nova Iorque, 1990.

Software

Durante este curso recorreu-se, frequentemente, ao programa

- *WinQSB: Software and Manual, Version 2.0*, 2001, 2002, (de Y.-Long Chang e K. Desai, publicado pela John Wiley & Sons, Inc.),

para resolver programas lineares e programas lineares inteiros ou inteiros mistos. O código permitiu, também, tirar proveito da estrutura de problemas de fluxo em redes (incluindo transportes, afectação, fluxo máximo e fluxo de custo mínimo sem restrições de capacidade).

Um dos códigos mais utilizados para resolver programas lineares chama-se CPLEX,

$$\texttt{http}://\texttt{www.ilog.com/products/cplex},$$

e permite correr métodos do tipo simplex e métodos de pontos interiores. Este código está preparado, também, para resolver problemas de programação linear inteira e tirar partido da estrutura dos problemas lineares de fluxo em redes.

Mais informação sobre *software* para as mais diversas classes de problemas de optimização pode ser encontrada nos guias:

- Decision Tree for Optimization Software:

$$\texttt{http}://\texttt{plato.la.asu.edu/guide.html},$$

- NEOS Guide:

$$\texttt{http}://\texttt{www}-\texttt{fp.mcs.anl.gov/otc/Guide}.$$