Métodos de Apoio à Decisão

Otimização Linear (Introdução)

João Pedro Pedroso

2023/2024

Formulação em programação matemática

Resolução de problemas:

- Utilização de GNU MathProg e do software `glpk`
  - linha de comando: `glpsol`
- Alternativa: software (comercial) `AMPL`
  - recentemente: *community edition*, gratuita para solvers open-source
- Versão com a licença para este curso:
  - `https://www.dcc.fc.up.pt/~jpp/AMPL/`

Uma fábrica de aços tem que decidir como utilizar o tempo da semana seguinte num moinho. O moinho utiliza restos de aço, podendo produzir fitas ou bobinas. As fitas podem ser produzidas à razão de 200 toneladas por hora, e as bobinas à razão de 140 toneladas por hora. Os lucros obtidos são de 25 contos por tonelada com as fitas e de 30 contos por tonelada com as bobinas. Atendendo à carteira de encomendas, a produção máxima na semana seguinte é de 6000 toneladas para fitas e de 4000 toneladas para bobinas.
Se nessa semana se dispuser de 40 horas de produção, quantas toneladas de cada um dos produtos deverão ser produzidas de forma a maximizar o lucro?

Formulação

$$
\begin{aligned}
\text{maximizar } z = \quad & 25x_F + 30x_B \\
\text{sujeito a} \quad & x_F/200 + x_B/140 && \leq 40 \\
& x_F && \leq 6000 \\
& x_B && \leq 4000 \\
& x_F, x_B && \geq 0
\end{aligned}
$$

Resolução

- GNU MathProg suporta um subconjunto da linguagem AMPL para otimização linear (com ou sem variáveis inteiras)
- AMPL é uma linguagem para problemas de otimização e programação por restrições
  - suporta também otimização não linear
  - utilizar um resolutor (*solver*) externo para obter a solução
- Modelo AMPL/GMLP Ficheiro `steel-a.mod`:

```
var XF;
var XB;

subject to Tempo: XF/200 + XB/140 <= 40;

subject to LimiteF: 0 <= XF <= 6000;

LimiteB: 0 <= XB <= 4000;  # "subject to" is optional

maximize lucro: 25*XF + 30*XB;
```

- *Try it:* `https://ampl.com/try-ampl/try-ampl-online/`

Resolução com GLPK

- Para resolver: na *shell* do Linux escrever:

```
glpsol --math steel-a.mod -o steel-a.sol
```

- A solução será registada no ficheiro `steel-a.sol`

```
Problem:    steel-a
Rows:       4
Columns:    2
Non-zeros:  6
Status:     OPTIMAL
Objective:  lucro = 192000 (MAXimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 Tempo        NU            40                         40          4200
     2 LimiteF      NU          6000             0          6000             4
     3 LimiteB      B           1400             0          4000
     4 lucro        B         192000

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 XF           B           6000
     2 XB           B           1400

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err. = 3.12e-16 on row 1
        max.rel.err. = 7.62e-18 on row 1
        High quality

KKT.PB: max.abs.err. = 0.00e+00 on row 0
        max.rel.err. = 0.00e+00 on row 0
```

AMPL: resolução do problema

- Para resolver de forma interativa em AMPL:

```
$ ampl
ampl: model steel-a.mod
ampl: solve;
MINOS 5.51: optimal solution found.
2 iterations, objective 192000
ampl: display XF, XB;
XF = 6000
XB = 1400
```

- Para automatizar: criar ficheiro `steel-a.run` contendo:

```
1  model steel-a.mod
2  solve;
3  display XF, XB;
```

  - depois, executar:

```
1  $ ampl steel-a.run
2  MINOS 5.51: optimal solution found.
3  2 iterations, objective 192000
4  XF = 6000
5  XB = 1400
6  $
```

GMPL: programas mais genéricos

Ficheiro `steel-b.mod`:

```
set PROD;

param rate {PROD} > 0;
param avail >= 0;
param profit {PROD};
param market {PROD} >= 0;

var Make {p in PROD} >= 0, <= market[p];

maximize total_profit:
    sum {p in PROD} profit[p] * Make[p];

subject to Time: sum {p in PROD} (1/rate[p]) * Make[p] <= avail;
```

GMPL: especificação dos dados

Ficheiro `steel-b.dat`:

```
set PROD := fitas bobinas;

param:     rate  profit  market :=
  fitas     200    25     6000
  bobinas   140    30     4000 ;

param avail := 40;
```

GLPK: resolução do problema

- Para resolver: na *shell* do Linux escrever:

```
glpsol --math steel-b.mod --data steel-b.dat -o steel-b.sol
```

- A solução será registada no ficheiro `steel-b.sol`

```
Problem:    steel-b
Rows:       2
Columns:    2
Non-zeros:  4
Status:     OPTIMAL
Objective:  total_profit = 192000 (MAXimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 total_profit B         192000
     2 Time         NU            40                          40          4200

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 Make[fitas]  NU          6000             0          6000             4
     2 Make[bobinas]
                    B           1400             0          4000

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err. = 3.12e-16 on row 2
        max.rel.err. = 7.62e-18 on row 2
        High quality

KKT.PB: max.abs.err. = 0.00e+00 on row 0
        max.rel.err. = 0.00e+00 on row 0
        High quality
```

AMPL: resolução do problema

- Ficheiro `[...].run`:

```
model steel-b.mod
data steel-b.dat
solve;
display Make;
```

  - depois, executar:

```
$ ampl steel-b.run
MINOS 5.51: optimal solution found.
2 iterations, objective 192000
Make [*] :=
bobinas  1400
  fitas  6000
;

$
```

Otimização linear (ou *programação linear*)

Otimização linear: definições prévias

Função linear A função $f(x_1, x_2, \ldots, x_n)$ é uma *função linear* se e só se para algum conjunto de constantes $c_1, c_2, \ldots, c_n$ se pode escrever

$$f(x_1, x_2, \ldots, x_n) = c_1\ x_1 + c_2\ x_2 + \ldots c_n\ x_n$$

Desigualdade linear Para qualquer função linear $f(x_1, x_2, \ldots, x_n)$ e qualquer número b, as desigualdades

$$f(x_1, x_2, \ldots, x_n) \leq b$$

$$f(x_1, x_2, \ldots, x_n) \geq b$$

são desigualdades lineares

Problema de otimização linear (PL)

- Problema linear é um problema de otimização para o qual se verifica o seguinte:
  - Tenta-se maximizar (ou minimizar) uma *função linear* das variáveis de decisão. À função que deverá ser maximizada ou minimizada chama-se *função objetivo*.
  - Os valores das variáveis de decisão deverão satisfazer um conjunto de *restrições*. Cada restrição deverá ser uma equação linear ou uma desigualdade linear.
  - Uma variável x_i poderá ter associada uma *restrição de sinal*, especificando que x_i é não negativo, ou que x_i é não positivo. Caso contrário, diz-se que x_i não tem o sinal restringido.
- Região admissível para um PL é o conjunto dos pontos que satisfazem todas as restrições.
- Solução ótima de um problema linear de maximização:
  - é um ponto na região admissível com o *maior valor* da função objetivo
  - para um problema de minimização: ponto na região admissível com o menor valor da função objetivo.

Hipóteses da otimização linear

- Proporcionalidade e aditividade: a contribuição de cada uma das variáveis de decisão é *proporcional ao valor dessa variável*, e *independente do valor das outras variáveis*
  - na função objetivo
  - no lado esquerdo de cada restrição
- Divisibilidade: as variáveis podem admitir valores fraccionários
- Certeza nos coeficientes: todos os coeficientes são conhecidos exatamente (de forma deterministica)

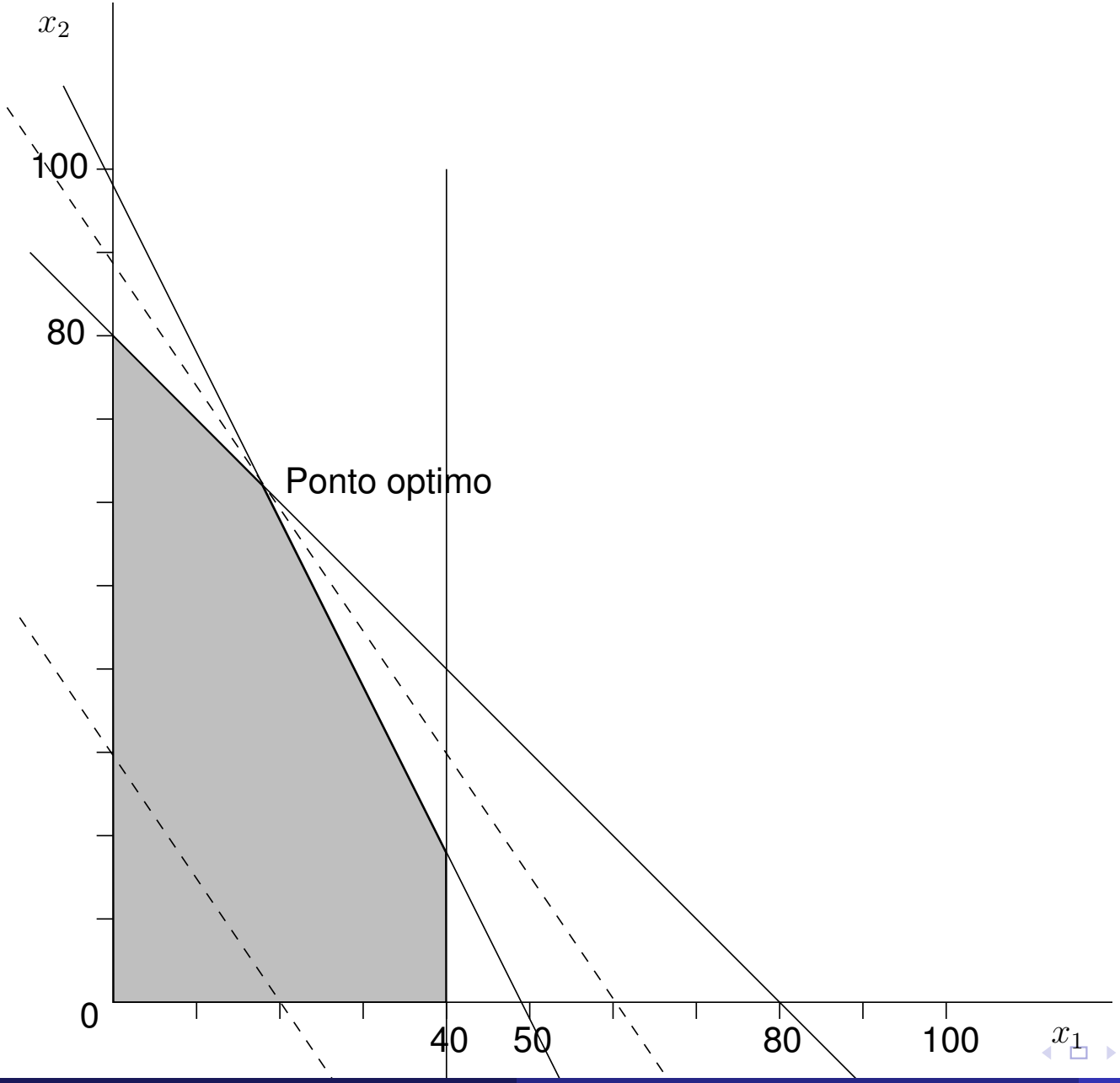
Resolução gráfica de problemas com duas variáveis

A companhia A. Madeira, Lda. fabrica dois tipos de móveis: mesas e cadeiras. Uma mesa vende-se por 27 contos, e usa 10 contos de materiais. As cadeiras vendem-se a 21 contos, e cada uma requer 9 contos de materiais.
Cada mesa construída aumenta os custos variáveis de trabalho e as despesas gerais em 14 contos; cada cadeira produzida aumenta estes custos em 10 contos.
A construção de mesas e cadeiras requer dois tipos de trabalho especializado: carpintaria e acabamentos. A produção de uma mesa requer 2 horas de acabamentos e uma hora de carpintaria; uma cadeira requer 1 hora de acabamentos e uma hora de carpintaria.
Em cada semana de trabalho, A. Madeira, Lda. pode obter todas as matérias primas que forem necessárias, mas tem disponíveis apenas 100 horas de mão de obra para acabamentos, e 80 horas de mão de obra de carpinteiros.
A procura de cadeiras é ilimitada, mas a venda de mesas é de, no máximo, 40 unidades por semana. A empresa pretende maximizar o lucro semanal.

Formulação em otimização linear

$$
\begin{array}{lll}
\text{maximizar} & z = 3x_1 + 2x_2 & \\
\text{sujeito a} \ : & 2x_1 + x_2 & \leq 100 \\
 & x_1 + x_2 & \leq 80 \\
 & x_1 & \leq 40 \\
 & x_1, x_2 \geq 0 &
\end{array}
$$

Resolução gráfica de problemas lineares com 2 variáveis



Terminologia

Soluções admissíveis e não admissíveis

- Uma solução admissível verifica todas as restrições
- Se uma solução não verificar alguma das restrições diz-se não admissível
- No exemplo anterior: soluções admissíveis encontram-se no espaço sombreado
- Ao conjunto de soluções admissíveis chama-se região admissível

Linhas de isocusto, linhas de isolucro

- Uma linha de isocusto (isolucro) é uma reta em que todos pontos têm o mesmo custo (lucro)
- Todas as linhas de isocusto/isolucro são paralelas entre si
- Para determinar o ponto ótimo, desloca-se uma linha de isocusto/isolucro paralelamente no sentido de diminuir os custos/aumentar os lucros
- O último ponto da região admissível a ser intersetado é o ótimo

Restrições ativas e não ativas

- Restrições ativas são aquelas que no ponto ótimo têm o termo linear, do lado esquerdo, tem o mesmo valor do termo independente (lado direito)
- Graficamente: são aquelas que intersetam o ponto ótimo

Conjuntos convexos, pontos extremos

- Um conjunto S é convexo se para qualquer par de pontos do conjunto, o segmento de reta que os une está completamente contido em S
- A região admissível de qualquer problema linear é um conjunto convexo
- Um ponto P diz-se um ponto extremo de um conjunto S se para qualquer segmento de reta que esteja completamente contido em S e que contenha P, se verifica que P é um ponto extremo desse segmento de reta

Noções estudadas

- Formulação em programação matemática
- Função e desigualdade lineares
- Problema de otimização linear
- Hipóteses da otimização linear
- Utilização do software glpk/ampl
- Resolução gráfica de problemas com duas variáveis.
- Noção de solução admissível e de solução não admissível.
- Linhas de isolucro/isocusto.
- Restrições ativas e não ativas.

Próxima aula

- Formulação em programação matemática: mais alguns exemplos.
- Casos especiais em otimização linear.