# Sumário

# Notação

- O problema de roteamento de veículos pode ser modelado num grafo orientado $G = (V, A)$, onde:
  - $V = \{0\} \cup N$.
    - 0 é o depósito; e
    - $N$ é o conjunto dos clientes, que têm procura/oferta.
  - $A$ é o conjunto dos arcos, que representam as ligações entre os clientes.
    - $A = \{(i, j) : i, j \in V \text{ e } i \neq j\}$
  - $d_i$ é a procura do cliente $i \in N$.
    - $D$ é a procura total, ou seja, $D = \sum_{i \in N} d_i$.
  - $K$ é número de veículos homogéneos disponíveis.
  - $Q$ é capacidade de cada veículo.
  - $c_{ij}$ é o custo de atravessar o arco $(i, j) \in A$.
- Queremos determinar o conjunto de circuitos de custo total mínimo que: (i) comecem e acabem no depósito; (ii) satisfaçam a procura de todos os clientes em $N$; e (iii) a procura total de cada rota não exceda a capacidade dos veículos.

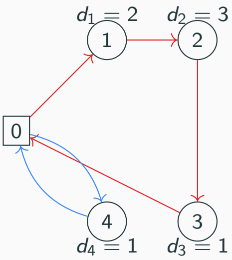
# Formulação genérica

$$x_{ij} = \begin{cases} 1, \text{se o arco } (i,j) \in A \text{ é usado em alguma das rotas} \\ 0, \text{caso contrário} \end{cases}$$

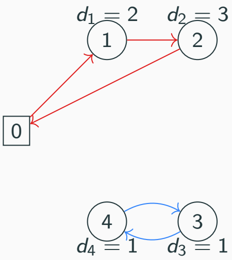
$$\begin{aligned}
\min \quad & \sum_{(i,j)\in A} c_{ij} x_{ij} \\
\text{sujeito a:} \quad & \sum_{i \in N} x_{0i} = K & & \text{(sai um arco do depósito para cada veículo)} \\
& \sum_{i \in V} x_{ji} = 1 & \forall j \in N \quad & \text{(um arco sai do cliente j)} \\
& \sum_{i \in V} x_{ij} = 1 & \forall j \in N \quad & \text{(um arco entra do cliente j)} \\
& \text{Não contém subcircuitos} \\
& \text{Satisfaz a capacidade dos veículos} \\
& x_{ij} \in \{0,1\} & \forall (i,j) \in A
\end{aligned}$$

# Formulação genérica

- Considerando $K = 2$ e $Q = 4$.




**(a)** Excede capacidade veículos.




**(b)** Contém subcircuitos.

**Figura 10:** Soluções não admissíveis para o VRP.

- Existem conjuntos de restrições que modelam as restrições de eliminação de subcircuitos e de capacidade.
  - Adaptações dos modelos de eliminação de subcircuitos para o TSP.

# Formulação MTZ

- Formulação de Miller-Tucker-Zemlin (MTZ)

$u_i$ - procura distribuída pelo veículo quando sai do nodo $i \in V \setminus \{0\}$.

$$
\begin{aligned}
\min \; & \sum_{(i,j)\in A} c_{ij}x_{ij} & & \\
\text{sujeito a:} \; & \sum_{i\in N} x_{0i} = K & & \text{(sai um arco do depósito para cada veículo)} \\
& \sum_{i\in V} x_{ji} = 1 & \forall j \in N \; & \text{(um arco sai do cliente j)} \\
& \sum_{i\in V} x_{ij} = 1 & \forall j \in N \; & \text{(um arco entra do cliente j)} \\
& u_i - u_j + Qx_{ij} \leq Q - d_j & \forall (i,j) \in A : i,j \neq 0 \; & \text{(a procura distribuída com que o veículo sai de j é a com que saiu de i mais o que distribuiu em j, se foi de i para j)} \\
& d_i \leq u_i \leq Q & \forall i \in N \; & \text{(a procura distribuída quando o veículo sai de i está entre a procura de i e a capacidade veículo)} \\
& x_{ij} \in \{0,1\} & \forall (i,j) \in A &
\end{aligned}
$$

- Consideremos $K = 2$ e $Q = 4$.

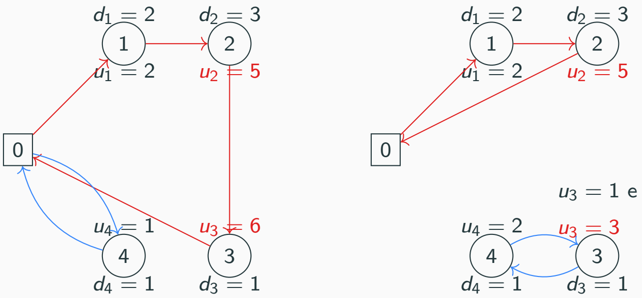



**Figura 11:** Soluções não admissíveis para o VRP.

- Considerando $K = 2$ e $Q = 4$.

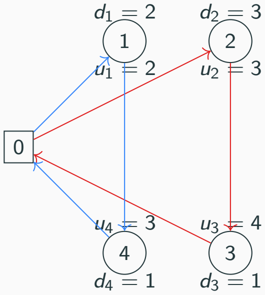



**Figura 12:** Solução admissível para o VRP.

# Formulação SCF

- Formulação de fluxo único - Single Commodity Flow (SCF)

$f_{ij}$ - fluxo que atravessa o arco $(i, j) \in A$, que corresponde à procura total por satisfazer quando o veículo atravessa arco $(i, j) \in A$.

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{i\in N} x_{0i} = K \qquad \text{(sai um arco do depósito para cada veículo)}$$

$$\sum_{i\in V} x_{ji} = 1 \qquad \forall j \in N \quad \text{(um arco sai do cliente j)}$$

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall j \in N \quad \text{(um arco entra do cliente j)}$$

$$\sum_{i\in V\setminus\{0\}} f_{0i} = D \qquad \text{(}D\text{ é a procura total por satisfazer quando os veículos saem do depósito)}$$

$$\sum_{j\in V:j\neq i} f_{ji} = \sum_{j\in V:j\neq i} f_{ij} + d_i \qquad \forall i \in N \quad \text{(procura por satisfazer quando entra em i}$$

é a com que sai de i mais a que foi satisfeita em i)

$$0 \leq f_{ij} \leq Q x_{ij} \qquad \forall (i, j) \in A \quad \text{(limites capacidade disponível)}$$

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in A$$

- Consideremos $K = 2$ e $Q = 4$.

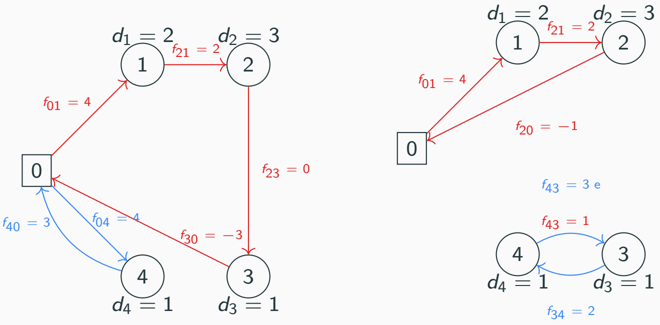



**Figura 13:** Soluções não admissíveis para o VRP.

- Consideremos $K = 2$ e $Q = 4$.

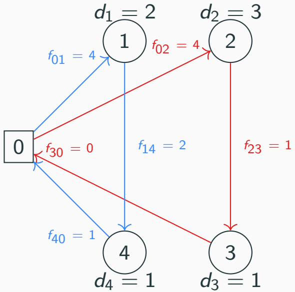



**Figura 14:** Solução admissível para o VRP.

## Formulação SCF

- Os limites das variáveis $f$ podem ser melhorados.
  - Quando saímos de $i$ qual é a procura máxima por satisfazer?
    - Como o $i$ já foi visitado, já lá ficaram $d_i$ unidades de fluxo. Logo, a procura por satisfazer é menor ou igual que $Q - d_i$.
  - Quando entramos em $j$ qual é a procura mínima por satisfazer?
    - Como o veículo ainda vai visitar o $j$, tem pelo menos a procura do $j$ para satisfazer. Logo, a procura por satisfazer é maior ou igual que $d_j$.

- Podemos substituir as restrições limites capacidade disponível por

$$d_j x_{ij} \leq f_{ij} \leq (Q - d_i) x_{ij}, \qquad \forall (i,j) \in A$$

Nota: Para simplificar a apresentação das restrições consideramos que $d_0 = 0$.

# Formulação RCC

- Formulação de cortes de capacidade arredondados - *Rounded capacity cuts* (RCC)[5]

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{i\in N} x_{0i} = K$$ (sai um arco do depósito para cada veículo)

$$\sum_{i\in V} x_{ji} = 1 \qquad \forall j \in N$$ (um arco sai do cliente j)

$$\sum_{i\in V} x_{ij} = 1 \qquad \forall j \in N$$ (um arco entra do cliente j)

$$\sum_{i\in V\setminus S} \sum_{j\in S} x_{ij} \geq \left\lceil \frac{\sum_{i\in S} d_i}{Q} \right\rceil \qquad \forall S \subset N : S \neq \emptyset$$ (o número de veículos necessários para satisfazer a procura de um subconjunto de clientes é a soma da procura a dividir pela capacidade do veículo arredondada para cima [variáveis inteiras])

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A$$

[5] Laporte, G., Mercure, H., & Nobert, Y. (1986). *An exact algorithm for the asymmetrical capacitated vehicle routing problem.* Networks, 16(1), 33-46.

# Formulação RCC

- As restrições $\sum_{i \in V \setminus S} \sum_{j \in S} x_{ij} \geq \left\lceil \frac{\sum_{i \in S} d_i}{Q} \right\rceil$ são em número exponencial.
  - Esta formulação deve ser resolvida com recurso a um algoritmo de *branch-and-cut*.
    - As restrições em número exponencial podem ser utilizadas como desigualdades válidas para as outras formulações apresentadas

# Formulação RCC

- Consideremos $K = 2$ e $Q = 4$.

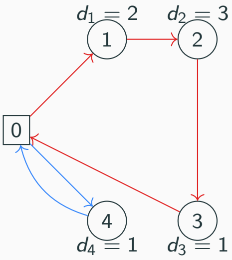



**Figura 15:** Solução não admissível para o VRP.

- Considere-se $S = \{1, 2, 3\}$. Temos $\sum_{i \in S} = d_1 + d_2 + d_3 = 2 + 3 + 1 = 6$ e $\lceil 6/4 \rceil = \lceil 1.50 \rceil = 2$ e apenas existe um arco com início em $V \setminus S$ e fim em $S$.
    - Um corte RCC para esta solução é $x_{01} + x_{02} + x_{03} + x_{41} + x_{42} + x_{43} \geq 2$.

# Formulação RCC

- Consideremos $K = 2$ e $Q = 4$.

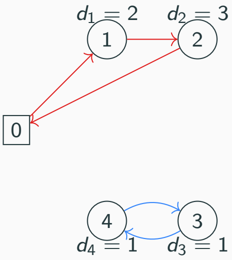



**Figura 16:** Solução não admissível para o VRP.

- Considere-se $S = \{3, 4\}$. Temos $\sum_{i \in S} = d_3 + d_4 = 1 + 1 = 2$ e $\lceil 2/4 \rceil = \lceil 0.50 \rceil = 1$ e não existe nenhum arco com início em $V \setminus S$ e fim em $S$ usado na solução.
  - Um corte RCC para esta solução é $x_{03} + x_{04} + x_{13} + x_{14} + x_{23} + x_{24} \geq 1$.

# Relaxações do problema do roteamento de veículos

- Qualquer formulação válida para o problema do roteamento de veículos garante que

  1. cada nodo tem um arco a entrar e um arco a sair;
  2. não existem subcircuitos; e
  3. a procura de cada rota não excede a capacidade do veículo.

$$
\begin{aligned}
& \min \sum_{(i,j)\in A} c_{ij} x_{ij} & & \\
\text{sujeito a: } & \sum_{i\in N} x_{0i} = K & & \text{(sai um arco do depósito para cada veículo)} \\
& \sum_{i\in V} x_{ji} = 1 & \forall j \in N & \quad \text{(um arco sai do cliente j)} \\
& \sum_{i\in V} x_{ij} = 1 & \forall j \in N & \quad \text{(um arco entra do cliente j)} \\
& \text{Não contém subcircuitos} & & \\
& \text{Satisfaz a capacidade dos veículos} & & \\
& x_{ij} \in \{0,1\} & \forall (i,j) \in A &
\end{aligned}
$$

# Relaxações do problema de roteamento de veículos

- Podemos obter relaxações removendo cada conjunto de restrições da formulação genérica, contudo
  - Se mantivermos o conjunto de restrições Não contém subcircuitos obtemos o problema de do caixeiro viajante múltiplo, que é um problema difícil.
  - Se mantivermos o conjunto de restrições Satisfaz a capacidade de veículos obtemos o problema de empacotamento, que é um problema difícil.

- Apenas vamos considerar a relaxação em que relaxamos ambos os conjuntos de restrições simultaneamente.

# Relaxações do problema de roteamento de veículos

- Relaxando as restrições de eliminação de subcircuitos e de capacidade de veículos obtemos:

$$
\begin{aligned}
(REL1) \equiv \min \sum_{(i,j)\in A} c_{ij}x_{ij} & & \\
\text{sujeito a: } \sum_{i\in N} x_{0i} = K & & \text{(sai um arco do depósito para cada veículo)} \\
\sum_{i\in V} x_{ji} = 1 & \quad \forall j \in N & \text{(um arco sai do cliente j)} \\
\sum_{i\in V} x_{ij} = 1 & \quad \forall j \in N & \text{(um arco entra do cliente j)} \\
x_{ij} \in \{0,1\} & \quad \forall (i,j) \in A &
\end{aligned}
$$

# Heurísticas para o problema do roteamento de veículos

- Heurísticas construtivas
  - Heurística de Clarke & Wright.
- Heurísticas de decomposição
  - Agrupar $\rightarrow$ Rotear
    1. Particionar do conjunto de clientes $N$ em $K$ subconjuntos que satisfaçam as restrições de capacidade.
    2. Determinar uma rota admissível em cada um dos subconjuntos $\rightarrow$ Resolver um TSP.
  - Rotear $\rightarrow$ Agrupar
    1. Determinar uma rota que inclui todos os clientes $\rightarrow$ Resolver um TSP.
    2. Dividir a rota em $K$ subrotas que satisfaçam as restrições de capacidade.

# Heurística de Clarke & Wright

- É uma heurística gulosa para o problema do roteamento de veículos[6].

## Heurística de Clarke & Wright

**for all** Cliente $i \in N$ **do**
    Calcular a rota $R_i = (0, i, 0)$.
**end for**
**for all** Par de clientes $i, j \in N : i \neq j$ **do**
    Calcular a poupança $s_{ij} = c_{i0} + c_{0j} - c_{ij}$.
**end for**
Fazer $R = \emptyset$ e $i \leftarrow 1$.
**while** A solução não é admissível **do**
    Seja $(i^*, j^*)$ o par que origina a $i$-ésima maior poupança.
    **if** $i^*$ e $j^*$ não são visitados **then**
        Criar a nova rota $(0, i^*, j^*, 0)$ se $d_{i^*} + d_{j^*} < Q$ e eliminar $R_{i^*}$ e $R_{j^*}$.
    **else if** Existem arcos $(0, i^*)$, $(0, j^*)$, $(i^*, 0)$ ou $(j^*, 0)$ em $R \cup \left( \cup_{i=1}^{|N|} R_i \right)$ **then**
        Juntar as rotas da seguinte forma $(0, \ldots, i^*, j^*, \ldots, 0)$ se a procura da nova rota não exceder $Q$.
    **end if**
    Fazer $i \leftarrow i + 1$.
**end while**

[6] Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations research, 12(4), 568-581.

# Exemplo

- Calcular a poupança para cada par de clientes.
  - Exemplo par $(1,2)$ : $s_{12} = c_{10} + c_{02} - c_{12} = 28 + 31 - 21 = 38$.

| $s_{ij}$ | A | B | C | D | E |
|---|---|---|---|---|---|
| A | - | 38 | 19 | 27 | 42 |
| B | - | - | 13 | 36 | 33 |
| C | - | - | - | 15 | 27 |
| D | - | - | - | - | 34 |

| | 0 | A | B | C | D | E |
|---|---|---|---|---|---|---|
| 0 | - | 28 | 31 | 20 | 25 | 34 |
| A | 28 | - | 21 | 29 | 26 | 20 |
| B | 31 | 21 | - | 38 | 20 | 32 |
| C | 20 | 29 | 38 | - | 30 | 27 |
| D | 25 | 26 | 20 | 30 | - | 25 |
| E | 34 | 20 | 32 | 27 | 25 | - |

| $Q = 100$ | A | B | C | D | E |
|---|---|---|---|---|---|
| $d_i$ | 37 | 35 | 30 | 25 | 32 |

Nota: Como a matriz de custos é simétrica temos $s_{ij} = s_{ji}$.

- Ordenar pares de clientes pela maior poupança: $(B,C)$, $(C,D)$, $(A,C)$, $(A,D)$, $(B,E)$, $(D,E)$, $(B,D)$, $(A,B)$ e $(A,E)$. Fazer $R = \emptyset$ e $i \leftarrow 1$.

- $R$ é admissível? Não.
  - Considere-se o par $(B,C)$.
  - Clientes $B$ e $C$ são não visitados? Sim.
    - Como $d_B + d_C = 69 < 100$, vamos criar uma nova rota. $\leftarrow$ $R = \{(0, B, C, 0)\}$.
  - $i \leftarrow i + 1 = 2$.

- $R$ é admissível? Não.
  - Considere-se o par $(C,D)$.
  - Clientes $C$ e $D$ não são visitados? Não.
  - Clientes $C$ ou $D$ estão adjacentes ao depósito no conjunto das rotas? Sim.
    - Como $d_B + d_C + d_D = 104 > 100$, não adicionamos um novo nodo à solução.
  - $i \leftarrow i + 1 = 3$.

# Exemplo

| | 0 | $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|---|---|
| 0 | - | 28 | 31 | 20 | 25 | 34 |
| $A$ | 28 | - | 21 | 29 | 26 | 20 |
| $B$ | 31 | 21 | - | 38 | 20 | 32 |
| $C$ | 20 | 29 | 38 | - | 30 | 27 |
| $D$ | 25 | 26 | 20 | 30 | - | 25 |
| $E$ | 34 | 20 | 32 | 27 | 25 | - |

| $Q = 100$ | $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|---|
| $d_i$ | 37 | 35 | 30 | 25 | 32 |

- $R$ é admissível? Não.
  - Considere-se o par $(A, C)$.
  - Clientes $A$ e $C$ não são visitados? Não.
  - Clientes $A$ ou $C$ estão adjacentes ao depósito no conjunto das rotas? Sim.
    - Como $d_B + d_C + d_A = 94 \leq 100$, vamos adicionar um novo nodo à rota em R já existente e eliminar a $R_A$. $\rightarrow$ $R = \{(0, B, C, A, 0)\}$.
  - $i \leftarrow i + 1 = 4$.
- $R$ é admissível? Não.
  - Considere-se o par $(A, D)$.
  - Clientes $A$ e $D$ não são visitados? Não. $\rightarrow$ Como ambos os clientes são visitados podemos saltar.
  - $i \leftarrow i + 1 = 5$.
- $R$ é admissível? Não.
  - Considere-se o par $(B, E)$.
  - Clientes $B$ e $E$ não são visitados? Não.
  - Clientes $B$ ou $E$ estão adjacentes ao depósito no conjunto das rotas? Sim.
    - Como $d_B + d_C + d_D + d_E = 104 > 100$, não vamos adicionar um novo nodo à solução.
  - $i \leftarrow i + 1 = 6$.

# Exemplo

- $R$ é admissível? Não.
  - Clientes $D$ e $E$ não são visitados? Sim.
    - Como $d_D + d_E = 69 < 100$, vamos criar uma nova rota. $\leftarrow R = \{(0, B, C, A, 0); (0, D, E, 0)\}$.
  - $i \leftarrow i + 1 = 7$.
- $R$ é admissível? Sim. FIM.

| | 0 | $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|---|---|
| 0 | - | 28 | 31 | 20 | 25 | 34 |
| $A$ | 28 | - | 21 | 29 | 26 | 20 |
| $B$ | 31 | 21 | - | 38 | 20 | 32 |
| $C$ | 20 | 29 | 38 | - | 30 | 27 |
| $D$ | 25 | 26 | 20 | 30 | - | 25 |
| $E$ | 34 | 20 | 32 | 27 | 25 | - |

| $Q = 100$ | $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|---|
| $d_i$ | 37 | 35 | 30 | 25 | 32 |

A solução obtida com a heurística do Clarke & Wright é $\{(0, B, C, A, 0); (0, D, E, 0)\}$, que tem valor $c_{0B} + c_{BC} + c_{CA} + c_{A0} + c_{0D} + c_{DE} + c_{E0} = 31 + 38 + 29 + 28 + 25 + 25 + 34 = 210$.

# Exemplo

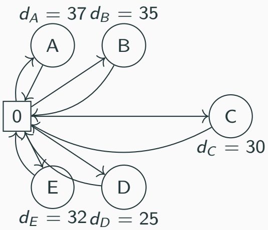



**Figura 17:** Inicialização.

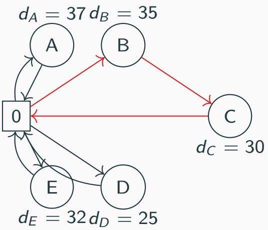



**Figura 18:** Iteração 1.

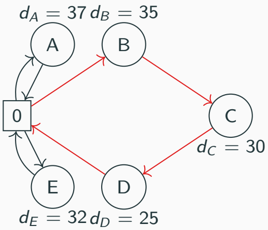



**Figura 19:** Iteração 3.

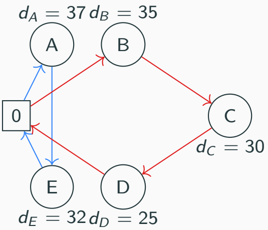



**Figura 20:** Iteração 7.

# VRP Spreadsheet Solver

- O VRP Spreadsheet Solver é uma ferramenta desenvolvida em Visual Basic for Applications (VBA) que permite obter soluções admissíveis para diversas variantes do problema de roteamento de veículos.
  - Os procedimentos implementados no VRP Solver são heurísticos. $\Longrightarrow$ Não podemos garantir que a solução obtida pela ferramenta seja a ótima.
- O download da ferramenta pode ser feito em
  - `https://people.bath.ac.uk/ge277/vrp-spreadsheet-solver/`
- Detalhes sobre a ferramenta podem ser encontrados em
  - Erdoğan, G. (2017). *An open source spreadsheet solver for vehicle routing problems*. Computers & Operations Research, 84, 62-72.

# Notas sobre o problema do roteamento de veículos

- Encontrar soluções admissíveis para o problema do roteamento de veículos pode ser um problema de difícil resolução dependendo da instância.
  - Resolver um problema de empacotamento.
- Existem inúmeras variantes do problema de roteamento de veículos, como por exemplo:
  - Problema do roteamento de veículos com janelas temporais.
  - Problema do roteamento de veículos com uma frota heterogénea.
  - Problema do roteamento de veículos com *backhauls*.
  - Problema do roteamento de veículos periódico.
  - Problema do roteamento de veículos com *split delivery*.

# Referências bibliográficas


- Drexl, M. (2012). *Rich Vehicle Routing in Theory and Practice, Logistics Research*, Volume 5, pp. 47-63 (DOI: 10.1007/s12159-012-0080-2).
- Erdoğan, G. (2017). *An open source spreadsheet solver for vehicle routing problems*. Computers & Operations Research, 84, 62-72.
- Gutin, G., & Punnen, A. P. (Eds.). (2006). *The Traveling Salesman Problem and its Variations*, Springer Science & Business Media, New York.
- Toth, P. & D. Vigo (2014). *Vehicle Routing Problems, Methods, and Application*, 2nd ed., MOS-SIAM Series on Optimization, Philadelphia.
- Wolsey, L. A. (1998). *Integer programming*. John Wiley & Sons.