Simulação e Otimização
Cap. 2

SIMO/MQDEE

MARIA CÂNDIDA MOURÃO

(cmourao@iseg.ulisboa.pt)

Programa

Cap. 1 – Técnicas de Resolução em Otimização Combinatória
- Relaxações
- Resolução exata de problemas
  - Algoritmo de *branch-and-bound*
  - Algoritmo de planos de corte
- Utilização de software

Cap. 2 – Problemas de Otimização Combinatória - Roteamento
- Problemas de roteamento nos nodos
- Problemas de roteamento nos arcos
- Utilização de Software – VRP **Spreadsheet Solver**

Cap. 3 – Modelos de Investigação Operacional em Simulação
- Simulação e otimização
- Geração de instâncias de problemas de otimização
- Utilização de software de simulação – SIMUL8

Bibliografia

- **Corberán, Á. & G. Laporte** (2014); *Arc Routing Problems, Methods, and Application*; MOS-SIAM Series on Optimization, Philadelphia.
- 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 and Operations Research, Volume 84, pp. 62-72 (DOI: 10.1016/j.cor.2017.02.022).
- **Hillier, F.S. & G.J. Lieberman** (2010), *Introduction to Operations Research*, 9th ed., McGraw-Hill, New York.
- Mourão, M.C. & L.S. Pinto (2017); *An updated annotated bibliography on arc routing problems*, Networks, Volume 70(3), pp. 144-194 (DOI :10.1002/net.21762)
- **Toth, P. & D. Vigo** (2014); *Vehicle Routing Problems, Methods, and Application*; 2nd ed., MOS-SIAM Series on Optimization, Philadelphia.

Cap. 2 **Problemas de Roteamento**

aula 5

SIMO/MQDEE

MARIA CÂNDIDA MOURÃO

(cmourao@iseg.utl.pt)

Cap. 2 Problemas de Otimização Combinatória

➢ Problemas de Otimização Combinatória

- a solução ótima é um subconjunto de um conjunto finito.
- Problemas que "poderiam ser resolvidos" por enumeração!
- Crescimento exponencial!

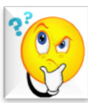
➢ Problemas de Roteamento

Cap. 2 Problemas de Otimização Combinatória

Metodologias

✓ Formulações

Modelos de redes & Modelos de PLI



✓ Heurísticas

básicas; construtivas; pesquisa local;
metaheurísticas;
matheurísticas

✓ Relaxações

✓ Softwares:

OPEN SOLVER - Formulações
VRP Spreadsheet SOLVER – Soluções Heurísticas
GERAÇÃO DE INSTÂNCIAS
www.tsp.gatech.edu

Cap. 2 Problemas de Otimização Combinatória

Problemas de Roteamento – Routing Problems (RP)

Identificar rotas que:

- Iniciam e terminam num mesmo ponto – depósito
- Minimizam a distância (tempo; custo; …) total percorrido

Roteamento nos Nodos
Node Routing Problem (NRP)

Roteamento nos Arcos
Arc Routing Problems (ARP)

LISBON SCHOOL OF ECONOMICS & MANAGEMENT UNIVERSIDADE DE LISBOA

Cap. 2 Problemas de Otimização Combinatória

Problemas de Roteamento nos Nodos – Node Routing Problems (NRP)

Características Genéricas dos problemas - identificar rotas que:

- Iniciam e terminam num mesmo ponto – depósito
- Minimizam a distância (tempo; custo; …) total percorrida
- Têm que passar por um conjunto de pontos (clientes) - todos ou só alguns
- Nos clientes pode ou não existir procura/oferta a satisfazer
- 1/vários veículos que podem ter capacidade limitada
- . . .

SIMULAÇÃO E OTIMIZAÇÃO (MQDEE) – 2017/18

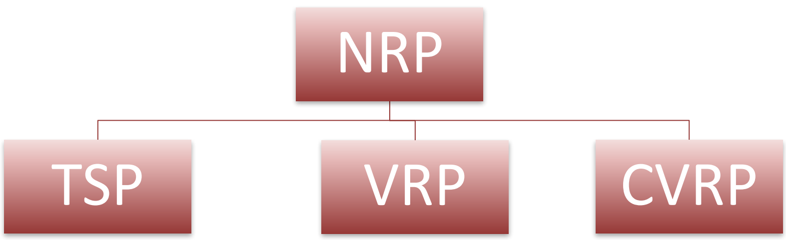
Node Routing Problems (NRP)

Outras especificidades:

- **Clientes:**
  - Time Windows – janelas temporais para satisfação de clientes
  - Restrições de precedência entre clientes
  - Clientes com oferta e/ou procura – Delivery & Pickup
  - Visitas periódicas
  - Pedido de veículo específico
  - . . .
- **Frota:**
  - um ou vários veículos
  - homogénea ou heterogénea
  - Custos fixos; variáveis; penalidades
  - Capacidade; Tempo limite
  - . . .
- **Objetivos:**
  - Min Custo
  - Min Nº veículos
  - Max. Proveitos
  - Um ou vários objetivos
  - . . .
- **Estrutura das Rotas:**
  - Fechadas ou abertas
  - Uma ou várias rotas por veículo
  - Rotas atrativas – equilibradas entre si; compactas; conexas; disjuntas
  - . . .
- **Âmbito do Planeamento:**
  - Horizonte temporal – tático; operacional; em tempo real
  - Tipo de dados – determinísticos; estocásticos; incertos
  - . . .

Drexl (2012)

Node Routing Problems (NRP)



Traveling Salesman Problem
(Problema do Caixeiro Viajante)

- 1 caixeiro
- Cidade de partida/chegada
- Visitar n cidades em tempo total mínimo

Vehicle Routing Problem

- 1 veículo
- Garagem de partida/chegada
- Visitar n clientes em tempo total mínimo

Capacitated VRP

- 1/K veículos de capacidade limitada
- Garagem de partida/chegada
- Satisfazer procura/oferta de n clientes em tempo total mínimo

Cap. 2 Problemas de Otimização Combinatória

Roteamento nos Nodos– NRP – em geral, NP-difíceis!

TSP – Dado um conjunto de cidades e as distâncias entre cada par de cidades, **qual** a rota de comprimento total mínimo que, partindo de uma cidade específica, passa em cada uma das restantes cidades exatamente uma vez e volta à cidade de partida?

- Nº SA → $(n-1)!$ (caso orientado)
- Enumeração -> só se conseguem resolver **instâncias** de pequenas dimensões!
- **Formulações; Minorantes; Majorantes**

n	$(n-1)!$	n	$(n-1)!$
5	24		
10	362’880	150	3.81×10^{260}
100	9.33×10^{155}	1001	4.02×10^{2567}

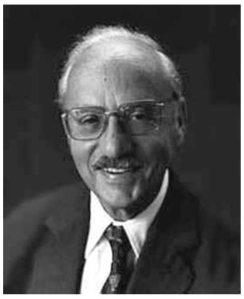
TSP – História

BC – Before Computers

AD – After Dantzig

- Matemático Russo;
- Método do Simplex
- Pai da PL & PLI

George Dantzig
1914 - 2005



George B. Dantzig developed such a method in 1947 while being assigned to the U.S. military. Ideally suited to computer use, the method is used routinely on applied problems involving hundreds and even thousands of variables and problem constraints.

A Short History of the TSP
G. Laporte;
CRT & GERAD HEC Montréal, Canada, Working Paper, 2006

TSP – História + Antiga

1759 - Euler ("Solution d'une question curieuse qui ne paraît soumise à aucune analyse"), Mémoire de l'Académie des Sciences de Berlin 15, 310–337 published in Opera Omnia (1) 7 (1766) 26–56. Studied the Knight's tour problem

1856 - Kirkman, rector of the parish of Craft with Southworth, Lancashire, from 1840 to 1892. Philosophical Transactions of the Royal Society, London, 146 (1856) 413–418. "Given the graph of a polyhedron, can one always final a circuit which passes through each vertex one and only once?"

1856 – Hamilton, Icosian game (marketed in 1859): find paths and circuits on the dodecahedral graph, satisfying certain conditions (e.g., adjacency conditions, etc.). The rights were sold for £25 to a wholesaler dealer in games and puzzles.

1930 – Menger, Hamiltonian path: "We call this the messenger problem (because in practice the problem has to be solved by every postman, and also by many travelers): finding the shortest path joining all of a finite set of points, whose pairwise distances are known".

1931-32 - Book published in German: "**The Traveling Salesman Problem**, how he should be and what he should do to be successful in his business. By a veteran traveling salesman". Last chapter: "By a proper choice and scheduling of the tour, one can often gain so much time that we have to make some suggestions . . . The most important aspect is to cover as many location as possible without visiting a location twice".

1937 ? - Tucker Introduced the problem to Flood in (1937) according to Flood, and in 1931–32 according to Tucker, in relation with a school-bus routing study in New Jersey.

A Short History of the TSP; G. Lporte;
CRT & GERAD HEC Montréal, Canada, Working Paper, 2006

TSP – História + Recente

TSP Simétrico

- Dantzig, Fulkerson & Johnson (1954) - origem do "branch-and-cut"

TSP Assimétrico

- Eastman's thesis (1958)
- Artigos de
  - Land and Doig (1960)
  - Little, Murty, Sweeney and Karel (1963), and the origin of

  Origem do "branch-and-bound"

A Short History of the TSP;
G. Lporte;
CRT & GERAD HEC Montréal, Canada, Working Paper, 2006

Heurísticas

- Artigos de Croes (1958) e de Lin (1965) – origem dos Alg. de Pesquisa Local

TSP – História

- Origem do TSP – desconhecida! 1832 – livro alemão – rotas dos caixeiros viajantes
- Nome: 1930-1940
- Evolução: desafios de problemas com
  - 49 cidades - USA - Dnatzig, 1954
  - 120 cidades – Alemanha – Grötschel, 1977
  - 532cidades – USA – Padberg & Rinaldi, 1987
  - 666 cidades - à volta do mundo – GrÖtschel & Holland, 1987
  - ...
  - 85’900 cidades – Cook, Espinoza & Goycoolea, 2006
  - “1,904,711-city world TSP”: a 0.112% do ótimo em 256.1 dias em 2003 (Math Prog. Series B)

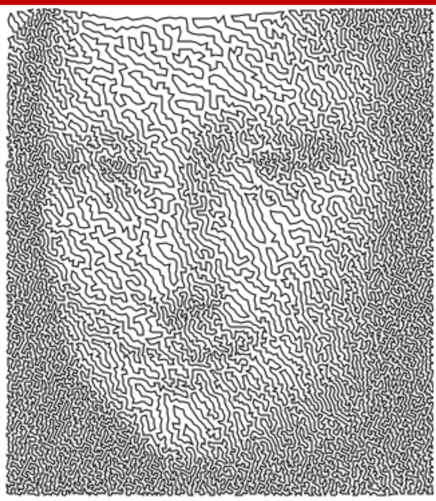
www.tsp.gatech.edu

TSPLIB
Desafios de hoje

The TSP – a computational study
D. Applegate, R. Bixby, V. Chvátal, W. Cook;
Princeton Series in Applied Mathematics, 2006, Oxford

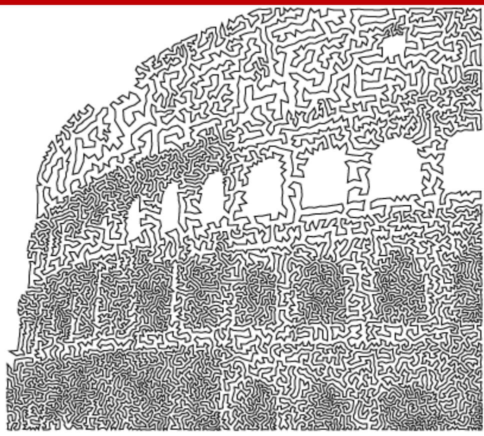
TSP - USA

TSP - curiosidades



10’000 cidades

Concorde Software – SAs!



11’999 cidades

C.S. Kaplan & R.Bosch, 2005
TSP Art, Technical Report, School of Computer Science, Univ of Waterloo, Canada

TSP - curiosidades

Mark Twain, *The Innocents Abroad, 1869*

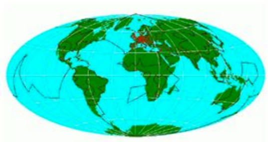
Proctor and Gamble ran a contest in 1962. The contest required solving a TSP on a specified 33 cities. There was a tie between many people who found the optimum.

Proctor and Gamble ran a contest in 1962. The contest required solving a TSP on a specified 33 cities. There was a tie between many people who found the optimum.

49 cities
• Dantzig, Fulkerson, and Johnson (1954)

120 cities
• Groetschel (1977).

• Padberg and Rinaldi (1987)
532 AT&T switch locations.



• Groetschel and Holland (1987)
tour of 666 interesting places.

• Padberg and Rinaldi (1987) tour through a layout of 2,392 that was obtained from Tektronics Incorporated.

• Applegate, Bixby, Chvátal, and Cook (1994) tour for a 7,397-city TSP that arose in a programmable logic array application at AT&T Bell Laboratories.

The traveling salesman problem on the WWW

- http://www.math.princeton.edu/tsp/index.html
- http://www.cs.rutgers.edu/~chvatal/tsp.html

Finite Mathematics, Feodor F. Dragan, Kent State University

TSP

Dados

- ✓ Rede Base: $G = (V, A \cup E)$

  - ✓ V – nodos; vértices – a visitar uma e uma só vez
  - ✓ Depósito ($0 \in V$) – indica o início e fim do percurso (cidade)
  - ✓ A – arcos (arcs) – ligações orientadas entre dois nodos
  - ✓ E – arestas (edges) – ligações não orientadas entre dois nodos
  - ✓ c_{ij} - custo (tempo; distância; ...) de deslocação entre $i \in V$ e $j \in V$

- ✓ Objetivo – Identificar um circuito em G, circuito que passa uma e uma só vez em cada nodo de G de custo total mínimo

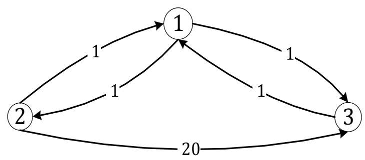
Casos: Orientado (assimétrico);
Não Orientado (simétrico);
Misto

TSP

- ✓ Circuito Hamiltoniano em G - circuito em G que passa uma e uma só vez em cada nodo de G

- ✓ Circuito Hamiltoniano de custo mínimo é sempre a SO do TSP?

- ✓ Desigualdade triangular falha – exemplo:



- ✓ Teorema: Se $\forall i, j \in V$ $c_{ij} \leq c_{iu} + c_{uj}$, $\forall u \neq i, u \neq j$, então um circuito Hamiltoniano de custo mínimo em G é uma SO do TSP, caso exista

J.R. Evans & E. Minieka, 1992
Optimization Algorithms for Networks and Graphs
2nd ed.Marcel Dekker, INC., NY

TSP – Heurísticas Construtivas

Logística e Redes…

H. Construtiva - Vizinho Mais Próximo (Nearest-Neighbor) – tipo *greedy*

1. Escolher um qualquer vértice inicial, $i \in V$;

   Iniciar a rota: $R = (i,)$, com $VR = \{i\} \subset \mathrm{V}$;

2. Procurar o vértice $j \in V \backslash \mathrm{VR}$ mais perto de i que ainda não esteja na rota, ou seja:

   $j = argmin\{c_{iv}: v \in V \backslash \mathrm{VR}\}$;

   Se $\nexists j$ **FIM** – não é possível encontrar uma SA por este método

   c.c. juntar o vértice j no final da rota R

3. **Se** $|VR| = |V|$ **FIM** $\Longrightarrow$ a rota R representa uma SA do TSP

   c.c. Fazer $i \leftarrow j$; Voltar a 2.

M.M. Flood, 1956
The TSP
Op. Res., 4, 61-75

- Problema: a escolha do próximo vértice tem tendência a deteriorar à medida que o algoritmo avança!
- Vizinho Mais Longe; Escolha aleatória numa certa vizinhança; ...

TSP – Heurísticas Construtivas

H. Construtiva: Inserção Mais Próxima (Nearest-Insertion)

1. Escolher um qualquer vértice inicial, $i \in V$; Fazer $\mathrm{VR} = \{i\}$

   Procurar o vértice $j \in V \backslash \mathrm{VR}$ mais perto de i;

   Iniciar a rota: $R = (i, j, i)$, com $VR = \{i, j\} \subset \mathrm{V}$;

2. Procurar o vértice $k \in V \backslash \mathrm{VR}$ mais perto de qualquer vértice $j \in VR$ e encontrar a aresta (v, u) que minimiza $c_{vk} + c_{ku} - c_{vu}$ $\forall v, u \in VR$;

   Se $\nexists k$ **FIM** – não é possível encontrar uma SA por este método

   c.c. juntar o vértice k entre v e j na rota R; atualizar VR

3. **Se** $|VR| = |V|$ **FIM** $\Longrightarrow$ a rota R representa uma SA do TSP

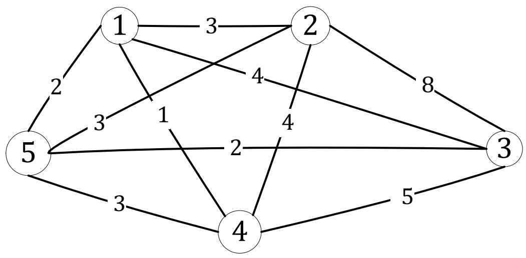
   c.c. Voltar a **2.**

D.J. Rosenkrantz, R.E. Sterns & P.M. Lewis, 1974
Approximate Algorithms for the TSP
Proc. 15th Ann. IEEE Symp. Switching and Automatic Theory, pp. 33-42

TSP – Heurísticas – exemplo 1

H. Construtiva - Inserção Mais Próxima

$$
\begin{array}{c|ccccc}
i,j & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & - & 3 & 4 & 1 & 2 \\
2 & & - & 8 & 4 & 3 \\
3 & & & - & 5 & 2 \\
4 & & & & - & 3 \\
5 & & & & & -
\end{array}
$$



TSP – Heurísticas – exemplo 1

H. Construtiva – Inserção Mais Próxima

R	VR	Nodo $\notin VR$ + Perto	custo	Posição	ΔCusto	Custo Total
1-4-1						

TSP – Heurísticas – exemplo 1

H. Construtiva – Inserção Mais Próxima

R	VR	Nodo $\notin VR$ + Perto	custo	Posição	ΔCusto	Custo Total
1-4-1	1	5	**2**	1-5-4	$\Delta = c_{15} + c_{54} - c_{14} = 2 + 3 - 1 = 4$	$2 + 4 = 6$
				4-5-1	Δ=4	
	4	5	3			
1-4-5-1	1	2	3			
	4	2	4			
	5	3	**2**	1-3-4	$\Delta = c_{13} + c_{34} - c_{14} = 4 + 5 - 1 = 8$	6+4=10
				4-3-5	$\Delta = c_{43} + c_{35} - c_{45} = 5 + 2 - 3 = 4$	
				5-3-1	$\Delta = c_{53} + c_{31} - c_{51} = 2 + 4 - 2 = 4$	

TSP – Heurísticas – exemplo 1

H. Construtiva – Inserção Mais Próxima

R	VR	Nodo $\notin VR$ + Perto	custo	Posição	ΔCusto	Custo Total
1-4-3-5-1	1					
	4					
	3					
	5					

TSP – Heurísticas – exemplo 1

H. Construtiva – Inserção Mais Próxima

R	VR	Nodo $\notin VR$ + Perto	custo	Posição	ΔCusto	Custo Total
1-4-3-5-1	1	2	3	1-2-4	$\Delta = c_{12} + c_{24} - c_{14} = 3 + 4 - 1 = 6$	10+4=14
				4-2-3	$\Delta = c_{42} + c_{23} - c_{43} = 4 + 8 - 5 = 7$	
				3-2-5	$\Delta = c_{32} + c_{25} - c_{35} = 8 + 3 - 2 = 9$	
				5-2-1	$\Delta = c_{52} + c_{21} - c_{51} = 3 + 3 - 2 = 4$	
	4	2	4			
	3	2	8			
	5	2	3			

Logo, $R = (1,4,3,5,2,1)$ de custo total 14

TSP – Heurísticas Melhorativas

H. Melhorativa – k-opt

1. Identificar S uma SA para o TSP
2. Substituir k arestas de S e outras k para gerar uma nova rota S'
3. **Se** o valor de S' é menor que o de S, fazer $S \leftarrow S'$ e voltar a 2.
   c.c., **FIM** S é a melhor SA

Logística e Redes…

S. Lin & B.W. Kernighan, 1973
Na Effective Heuristic Algorithm for the TSP
Op. Res., 21(2) pp. 498-516

TSP – Heurísticas Melhorativas

H. Melhorativa – Inversão de Sub-rotas

1. Identificar S uma SA para o TSP
2. Considerar todas as possibilidades de inverter subrotas (sem considerar toda a rota inicial) em S
3. **Se** nenhuma inversão melhora o valor da AS corrente S **FIM**

   c.c., Escolher a inversão que origina maior decréscimo no valor da SA para representar a nova SA;

   Atualizar S considerando a inversão escolhida e voltar a 2.

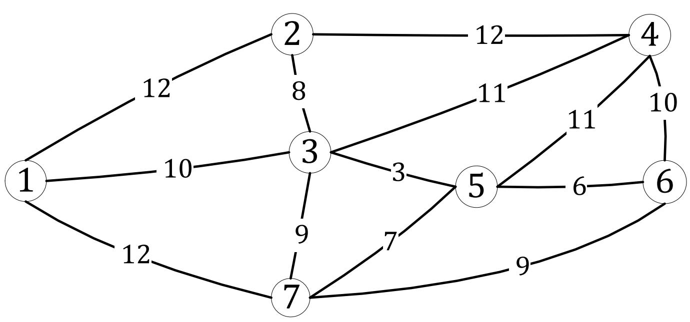
Caso Não Orientado

J. Little, K. Murty, D. Sweeney, C. Karel, 1963
An Algorithm for the TSP
Op. Res., 11(6), 972-989

TSP – Heurísticas – exemplo 2

LISBON SCHOOL OF ECONOMICS & MANAGEMENT
UNIVERSIDADE DE LISBOA

H. Melhorativa – Inversão de Sub-rotas



Seja $S = (1,2,3,4,5,6,7,1)$

TSP – Heurísticas – exemplo 2

H. Melhorativa – Inversão de Sub-rotas

Inverter	Rota	alteração	Distância total
S (inicial)	1-2-3-4-5-6-7-1		69

TSP – Heurísticas – exemplo 2

H. Melhorativa – Inversão de Sub-rotas

Inverter	Rota	alteração	Distância total
S (inicial)	1-2-3-4-5-6-7-1		69
2-3	1-3-2-4-5-6-7-1	$c_{13} + c_{24} - c_{12} - c_{34} = 10 + 12 - 12 - 11 = -1$	68
3-4	1-2-4-3-5-6-7-1	$c_{24} + c_{35} - c_{23} - c_{45} = 12 + 3 - 8 - 11 = -4$	65
4-5	1-2-3-5-4-6-7-1	$c_{35} + c_{46} - c_{34} - c_{56} = 3 + 10 - 11 - 6 = -4$	65
5-6	1-2-3-4-6-5-7-1	$c_{46} + c_{57} - c_{45} - c_{67} = 10 + 7 - 11 - 9 = -3$	66

TSP – Heurísticas – exemplo 2

H. Melhorativa – Inversão de Sub-rotas

Inverter	Rota	alteração	Distância total
S (inicial)	1-2-4-3-5-6-7-1		65

TSP – Heurísticas – exemplo 2

H. Melhorativa – Inversão de Sub-rotas

Inverter	Rota	alteração	Distância total
S (inicial)	1-2-4-3-5-6-7-1		65
2-4-3	1-3-4-2-5-6-7-1	não admissível	
4-3-5	1-2-5-3-4-6-7-1	não admissível	
3-5-6	1-2-4-6-5-3-7-1	$c_{46}+c_{37}-c_{43}-c_{67}$ $=10+9-11-9$ $=-1$	64
5-6-7	1-2-4-3-7-6-5-1	não admissível	

SIMULAÇÃO E OTIMIZAÇÃO (MQDEE) – 2017/18

TSP – Heurísticas – exemplo 2

H. Melhorativa – Inversão de Sub-rotas

Inverter	Rota	alteração	Distância total
S (inicial)	1-2-4-6-5-3-7-1		64

TSP – Heurísticas – exemplo 2

H. Melhorativa – Inversão de Sub-rotas

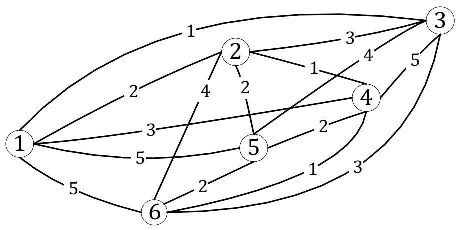
Inverter	Rota	alteração	Distância total
S (inicial)	1-2-4-6-5-3-7-1		64
2-4-6-5	1-5-6-4-2-3-7-1	não admissível	
4-6-5-3	1-2-3-5-6-4-7-1	não admissível	
6-5-3-7	1-2-4-7-3-5-6-1	não admissível	

Logo, $S = (1,2,4,6,5,3,7,1)$ de valor 64

TSP – Heurísticas – Exercícios

Considere a seguinte rede e matriz de custos:

$i \backslash j$	1	2	3	4	5	6
1	–	2	1	3	5	5
2		–	3	1	2	4
3			–	5	4	3
4				–	2	1
5					–	2
6						–



a) Identifique uma SA utilizando o algoritmo de inserção mais próxima

b) Utilizando o algoritmo de inversão de subrotas, determine uma SA, considerando a seguinte SA inicial:

1. 1-2-3-4-5-6-1
2. obtida em a)

Cap. 2 Problemas de Otimização Combinatória
aula 6

SIMO/MQDEE

Maria Cândida Mourão

(cmourao@iseg.utl.pt)

TSP

Cap. 2 – Problemas de Otimização Combinatória - Roteamento

Problemas de roteamento nos nodos

Problemas de roteamento nos arcos

Utilização de Software – VRP **Spreadsheet Solver**

- Roteamento nos nodos
  - Modelos – Casos Orientado e não Orientado
  - Relaxações
  - Solver/Excel

TSP –Modelos – caso orientado (assimétrico)

- Dados: $G = (V, A)$; c_{ij}
- Variáveis: $x_{ij} = \begin{cases} 1 & \text{se o caixeiro viaja de } i \in V \text{ para } j \in V \\ 0 & \text{c. c.} \end{cases}$
- Modelo:

$$(\text{P1}) \quad \min \sum_{(i,j)\in A} c_{ij}\, x_{ij}$$

Minimização do custo total

$$\text{s. a:} \begin{cases} \displaystyle\sum_{j:(i,j)\in A} x_{ij} = 1 & \forall i \in V \\ \displaystyle\sum_{k:(k,i)\in A} x_{ki} = 1 & \forall i \in V \\ \displaystyle\sum_{i\in S}\sum_{j\in V\setminus S} x_{ij} \geq 1 & \forall S \subset V \\ x_{ij} \in \{0,1\} & \forall (i,j) \in A \end{cases}$$

De cada nodo i só sai uma ligação

Só uma ligação é usada para entrar em cada nodo i

Restrições de eliminação de subcircuitos ilegais

TSP – Modelos – CO – exemplo 3

Considere a matriz de custos seguinte relativa a um problema de Caixeiro viajante

$$\begin{array}{c|cccc} i,j & 1 & 2 & 3 & 4 \\ \hline 1 & \infty & 3 & 5 & 1 \\ 2 & \infty & \infty & 1 & 6 \\ 3 & 6 & 1 & \infty & 3 \\ 4 & 1 & 5 & \infty & \infty \end{array}$$

a) Formalize o problema e resolva-o pelo Solver/Excel

b) Utilize o Solver para obter a solução do problema relaxado em que não se consideram as restrições de eliminação de subcircuitos

c) Introduza na solução de b) restrições de eliminação de subcircuitos até encontrar uma SO do TSP

d) Identifique uma SA utilizando a heurística do vizinho mais próximo e uma pela de melhor inserção

e) Compare os valores obtidos por todos os métodos

TSP –Modelos – caso NO (simétrico)

- Dados: $G = (V, E)$; c_e; $\delta(i)$ = conjunto de arestas incidentes em $i \in V$
- Variáveis: $x_e = \begin{cases} 1 & \text{se o caixeiro utiliza a aresta } e \in E \\ 0 & \text{c. c.} \end{cases}$
- Modelo:

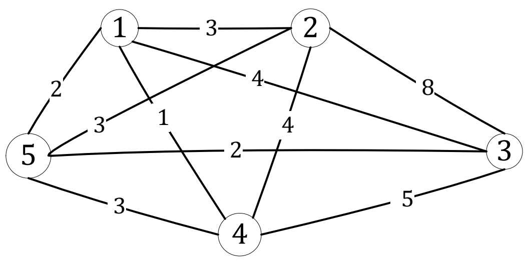
$$\text{(P2)} \quad \min \sum_{e \in E} c_e x_e$$ Minimização do custo total

$$\text{s. a:} \begin{cases} \sum_{e \in E} x_e = n & \text{São usadas } n \text{ arestas na solução} \\ \sum_{e \in \delta(i)} x_e = 2 \quad \forall i \in V & \text{Em cada nodo } i \text{ incidem } 2 \text{ arestas} \\ \sum_{e \in (S,T)} x_e \geq 1 \quad \forall (S,T), S \subset V, T = V \backslash S & \text{Restrições de eliminação de subcircuitos ilegais} \\ x_e \in \{0,1\} \quad \forall e \in E \end{cases}$$

TSP – Modelos – CNO - exemplo 1

Considere a matriz de custos seguinte relativa a um problema de Caixeiro viajante

$$\begin{array}{c|c}
i,j & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} - & 3 & 4 & 1 & 2 \\ & - & 8 & 4 & 3 \\ & & - & 5 & 2 \\ & & & - & 3 \\ & & & & - \end{pmatrix}
\end{array}$$



a) Formalize o problema

b) Utilize o Solver para obter a solução do problema relaxado em que não se consideram as restrições de eliminação de subcircuitos. Classifique a solução obtida.

c) Compare os valores da solução obtida com o majorante obtido pela H. de Inserção mais próxima.

TSP –Relaxações – caso O (assimétrico)

- ∀ SA verifica:
  - Cada nodo tem um arco a "entrar" e um a "sair"!
  - O circuito é conexo
- Relaxação:

~~Restrições de eliminação de subcircuitos ilegais~~

$$(\text{P1R1}) \quad \min \sum_{(i,j)\in A} c_{ij}\, x_{ij}$$ Minimização do custo total

$$\text{s. a:} \begin{cases} \sum_{j:(i,j)\in A} x_{ij} = 1 & \forall i \in V \\ \sum_{k:(k,i)\in A} x_{ki} = 1 & \forall i \in V \\ x_{ij} \in \{0,1\} & \forall (i,j) \in A \end{cases}$$

De cada nodo i só sai uma ligação

Só uma ligação é usada para entrar em cada nodo i

Problema de Afetação (PA)

TSP –Relaxações – caso 0 (assimétrico)

LISBON SCHOOL OF ECONOMICS & MANAGEMENT UNIVERSIDADE DE LISBOA

Resolução do PA:

- Construir um grafo bipartido: $G' = (S, T, A')$:
  - Para cada vértice $v \in V$ criar 2 vértices: $v_s \in S$ e $v_t \in T$
  - Para cada arco $(i, j) \in A$ criar um arco $(i_s, j_t) \in A'$ com custo c_{ij}
- Resolver o Problema de Afetação em G'
- Na solução do PA de custo mínimo em G' todos os vértices terão um arco a "sair" e um a "entrar"!
- Também se pode aplicar a problemas simétricos!

TSP –Relaxações – exemplos

LISBON SCHOOL OF ECONOMICS & MANAGEMENT UNIVERSIDADE DE LISBOA

a) Defina o PA que representa uma relaxação para o exemplo 3 (CO)

b) Resolva o problema de a) pelo Solver/Excel e, se necessário, introduzir cortes (quebras de subcircuitos ilegais) para encontrar a SO do problema inicial.

c) Defina o PA que representa uma relaxação para o exemplo 1 (CNO)

d) Resolva o problema de c) pelo Solver/Excel e, se necessário, introduzir cortes (quebras de subcircuitos ilegais) para encontrar a SO do problema inicial.

TSP –Relaxações – CO

- ∀ SA verifica:
  - Cada nodo tem um arco a "entrar" e um a "sair" !
  - O ciclo é conexo
- Relaxação – Sem restrições que garantem 1 arco a sair de cada nodo

$$(\text{P1R2}) \quad \min \sum_{(i,j)\in A} c_{ij}\, x_{ij}$$ Minimização do custo total

$$\text{s. a:} \begin{cases} \displaystyle\sum_{j:(j,i)\in A} x_{ji} = 1 \quad \forall i \in V \\ \displaystyle\sum_{i\in S}\sum_{j\in V\setminus S} x_{ij} \geq 1 \quad \forall S \subset V \\ x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \end{cases}$$

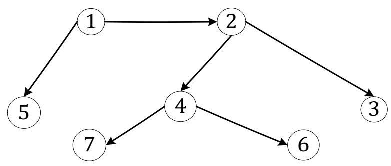
Só uma ligação é usada para entrar em cada nodo i

Restrições de eliminação de subcircuitos ilegais

∀ SA - 1 arco a "entrar" em cada nodo, num grafo conexo

TSP –Relaxações – Arborescência

- Árvore - grafo conexo s sem ciclos
- Arborescência - árvore com no máximo um arco a entrar a cada vértice, sendo a raiz o único vértice sem arcos a entrar!



- Arborescência geradora de G – arborescência que inclui todos os vértices de G
- Uma arborescência geradora de G com raiz num qualquer vértice de G, seja x, e em que se junta um arco a entrar em x, é um grafo conexo, em que todos os vértices têm um arco a entrar, mas em que podem existir vértices sem arcos a sair – Relaxação (P1R2)

TSP –Relaxações – Arborescência

Algoritmo – Relaxação:

- Identificar uma arborescência, T, de custo mínimo em G com raíz num qualquer vértice x
- Juntar a T o arco de menor custo a entrar em x

- O valor de T é um minorante para o valor do problema!

TSP –Relaxações – exemplos

a) Defina, a partir de uma arborescência geradora mínima, uma relaxação para o exemplo 3 e determine o correspondente minorante.

b) Compare o valor de a) com o minorante calculado pela resolução do PA.

c) Resolver o problema de a) pelo Solver/Excel e, se necessário, introduzir cortes (impondo que o grau externo dos vértices seja 1) para encontrar a SO do problema inicial.

TSP –Relaxações – CNO

- ∀ SA verifica:
  - Cada nodo tem e arestas nele incidentes
  - O ciclo é conexo
- Relaxação – Sem restrições que garantem que o grau de cada nodo é 2

$$(\text{P2R2})\ \min \sum_{e \in E} c_e\, x_e$$ Minimização do custo total

$$\text{s.a:} \begin{cases} \sum_{e \in E} x_e = n \\ \sum_{e \in (S,T)} x_e \geq 1 \quad \forall (S,T), S \subset V, T = V \setminus S \\ x_e \in \{0,1\} \quad \forall e \in A \end{cases}$$

Usadas n arestas

Restrições de eliminação de subcircuitos ilegais

∀ SA – n arestas que formam um grafo conexo

TSP –Relaxações – CNO

Algoritmo – Relaxação:

- Remover um vértice qualquer x de G
- Identificar uma árvore geradora no gafo resultante – tem $n-2$ arestas
- Juntar à árvore as duas arestas de menor custo incidentes em x

- O conjunto (grafo) construído satisfaz as restrições de (P2R2)!

é uma Árvore-1

TSP –Relaxações – exemplos

a) Definir, a partir de uma àrvore-1, uma relaxação para o exemplo 1 e determinar o correspondente minorante.

b) Comparar o valor de a) com o das restantes relaxações encontradas.

c) Resolver o problema de a) pelo Solver/Excel e, se necessário, introduzir cortes (impondo que o grau dos vértices seja 2) para encontrar a SO do problema inicial.

d) Comparar os valores dos minorantes obtidos com os valores das soluções heurísticas

TSP – Resumo

- Heurísticas
  - **H. Construtiva**: Vizinho + Perto; Inserção + Próxima
  - **H. Melhorativa**: k-opt; Inversão de subrotas
- Modelos
  - Caso Orientado
  - Caso Não Orientado
- Relaxações
  - SEM Rest. Subcircuitos ilegais ➡ PA
  - CO: SEM Rest. $\delta^+(i) = 1, \forall i$ ➡ Arborescência
  - CNO: SEM Rest. $\delta(i) = 2, \forall i$ ➡ Árvore-1

TSP – exercício - CNO

Considere a matriz de custos seguinte relativa a um problema de Caixeiro Viajante

	C1	C2	C3	C4	C5	C6
C1	∞	7	10	12	2	9
C2		∞	11	14	8	1
C3			∞	12	6	8
C4				∞	7	9
C5					∞	11

a) Formalize o problema.

b) Identifique uma solução admissível pelo método de inserção mais próxima.

c) Identifique um minorante recorrendo à resolução de um PA.

d) Introduza, na solução de c) restrições relaxadas até encontrar uma SO do TSP.

e) Repita b) e c) considerando a relaxação que resulta na resolução de um problema de árvore-1.

TSP – exercícios - CO

Considere a matriz de custos seguinte relativa a um problema de Caixeiro Viajante

	C1	C2	C3	C4	C5
C1	∞	3	7	6	2
C2	4	∞	11	9	6
C3	2	6	∞	1	5
C4	5	5	3	∞	4
C5	6	2	7	6	∞

a) Formalize o problema

b) Identifique uma SA pelo método de Inserção Mais Próxima

c) Identifique um minorante recorrendo à resolução de um PA.

d) Introduza, na solução de c) restrições relaxadas até encontrar uma SO do TSP.

e) Repita b) e c) considerando a relaxação que resulta na resolução de um problema de arborescência.

Cap. 2 Problemas de Otimização Combinatória
aula 7

SIMO/MQDEE

MARIA CÂNDIDA MOURÃO

(cmourao@iseg.utl.pt)

TSP – exercícios

Correção dos exercícios da aula passada

- ✓ Exemplo 1 – slide 55
- ✓ Exercício caso NO
- ✓ Exercício CO

SOLVER

TSP

Cap. 2 – Problemas de Otimização Combinatória - Roteamento

Problemas de roteamento nos nodos

Problemas de roteamento nos arcos

Utilização de Software – VRP **Spreadsheet Solver**

- VRP
  - Modelos – Casos Orientado e não Orientado
  - Relaxações
  - Heurística
  - Solver/Excel

P. Toth & D. Vigo (2014)
Vehicle Routing Problems, Methods, and Application
2nd ed., MOS-SIAM Series on Optimization, Philadelphia

SIMULAÇÃO E OTIMIZAÇÃO (MQDEE) – 2017/18 61

Node Routing Problems (NRP)

NRP

TSP — VRP — CVRP

TSP
Traveling Salesman Problem
(Problema do Caixeiro Viajante)

- 1 caixeiro
- Cidade de partida/chegada
- Visitar n cidades em tempo total mínimo

VRP
Vehicle Routing Problem

- 1 veículo
- Garagem de partida/chegada
- Visitar n clientes em tempo total mínimo

CVRP
Capacitated VRP

- 1/K veículos de capacidade limitada
- Garagem de partida/chegada
- Satisfazer procura/oferta de n clientes em tempo total mínimo

SIMULAÇÃO E OTIMIZAÇÃO (MQDEE) – 2017/18 62

VRP – Componentes Básicas

- Características da frota de veículos
- Condições de transporte
- Requisitos dos clientes – como podem ser satisfeitos
- Conceito de rota admissível
- Especificidades das rotas
- Dados a considerar – que custos e/ou proveitos; tempos; ofertas/procuras
- Horizonte de planeamento
- Sistemas dinâmicos – comunicação constante condutor/planeamento

VRP – Curiosidades

- 1959 – Dantzig & Ramser – *The Truck Dispaching Problem*

  Management Science, Vol. 6, pp. 80-91
  - Distribuição de gasolina
  - 1º modelo matemático
  - 1ª Heurística

- 1964 – Clarke & Wright – *Scheduling of vehicles from a central depot to a number of delivery points*

  Operations Research, Vol. 12, pp. 568-581
- Savings Heuristic – greedy

- Grupos de Investigadores
  - USA – Transportation Science and Logistics society
  - Europa – VeRoLog – Vehicle Routing and Logistics Optimization (www.verolog.eu)

VRP – características

Capacitated VRP

Gilbert Laporte
HEC Montréal

- Veículos de capacidade limitada
- Garagem de partida/chegada – depósito
- Satisfazer procura/oferta de n clientes em tempo (custo; distância; ...) total mínimo
- Decidindo:
  - que veículo deve satisfazer que clientes

  &

  - qual a sequência de clientes a visitar por cada veículo
- VRP - Componentes básicas Identificar – requisitos da procura

Em geral, NP-difíceis!

Capacitated VRP (CVRP) –Notação

- $N = \{1,2, \ldots, n\}$ – Clientes
- $V = N \cup \{0\}$; 0 – depósito
- Grafo:
  - $\mathrm{G} = (V, E)$ – Não Orientado $(\{i,j\} \in E)$ (grafo completo: $|E| = \frac{n(n+1)}{2}$)

  ou

  - $\mathrm{G} = (V, A)$ – Orientado $((i,j) \in A)$ (grafo completo: $|A| = n(n+1)$)
- $K = \{1, \ldots, |K|\}$ ($|K| \geq 1$) veículos homogéneos de capacidade limitada
- $Q > 0$ – capacidade de cada veículo
- c_{ij} – custo (distância; tempo;...) da ligação (i,j)
- q_i – procura/oferta do cliente $i \in N$
- $\delta^+(i)$ – nº de arcos a sair de $i \in V$; $\delta^-(i)$ – nº de arcos a entrar em $i \in V$
- $\delta^+(S)$– nº de arcos a sair de $S \subset V$; $\delta^-(S)$ – nº de arcos a entrar em$S \subset V$
- $\delta(i)$ – nº de arestas incidentes em $i \in V$; $\delta(S)$ – nº de arestas incidentes em $S \subset V$

Capacitated VRP (CVRP) – Notação

- *Rota* – sequência de vértices, com início e fim no depósito, $r = (0, i_1, i_2, \dots, i_s, 0)$ que visita um subconjunto de clientes, $S = \{i_1, i_2, \dots, i_s\} \subseteq N$
- A *rota* r é *admissível*:
  - se a procura dos clientes visitados não excede a capacidade do veículo, ou seja, se $q(S) = \sum_{i \in S} q_i \leq Q$

  e se
  - nenhum cliente é visitado mais de uma vez
- SA para o CVRP - $|K|$ rotas admissíveis, uma para cada veículo $k \in K$
- As rotas $r_1, r_2, \dots, r_{|K|}$ e os correspondentes conjuntos de clientes $S_1, S_2, \dots, S_{|K|}$ representam uma SA para o CVRP se todas as rotas forem admissíveis e se $S_1, S_2, \dots, S_{|K|}$ representarem uma partição de N

CVRP – Resolução

- Tarefas interligadas para identificar uma SA para o VRP:
  - Identificar uma partição do conjunto de clientes N em $|K|$ subconjuntos
  - Identificar uma rota admissível que sirva cada um dos subconjuntos de clientes - TSP
- Se resolvidas em separado não podemos garantir a otimalidade da solução!

CVRP – Modelos - CO

G. Laporte, H. Mercure, Y. Nobert, 1986
An exact algorithm for the assymetrical capacitated VRP
Networks, Vol. 16, pp. 33-46

- Dados: $G = (V, A)$; c_{ij}; q_i; K; Q;
- Variáveis: $x_{ij} = \begin{cases} 1 & \text{se 1 veículo viaja de } i \in V \text{ para } j \in V \\ 0 & \text{c.c.} \end{cases}$
- Modelo: $\boxed{|A| = n(n+1) \text{ variáveis!}}$

$$(\text{VRP1}) \quad \min \sum_{(i,j)\in A} c_{ij}\, x_{ij}$$
Minimização do custo total

$$\text{s.a:} \begin{cases} \displaystyle\sum_{j\in\delta^+(i)} x_{ij} = 1 \quad \forall i \in N & \text{Sai 1 vez de cada cliente } i \\ \displaystyle\sum_{k\in\delta^-(i)} x_{ki} = 1 \quad \forall i \in N & \text{Chega 1 vez a cada cliente } i \\ \displaystyle\sum_{j\in\delta^+(0)} x_{0j} = |K| & \text{São usados os } |K| \text{ veículos} \\ \displaystyle\sum_{j\in\delta^+(S)} x_{ij} \geq \left\lceil \frac{q(S)}{Q} \right\rceil \quad \forall S \subseteq N, S \neq \emptyset & \text{Restrições de eliminação de subcircuitos ilegais \& capacidade} \\ x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \end{cases}$$

VRP – Modelos – CO – exemplo 4

Considere a matriz de custos seguinte relativa a um problema de roteamento nos nodos

	Dep	C1	C2	C3	C4	C5	C6	C7
Dep	-	20	55	50	40	15	30	40
C1	25	-	40	20	50	25	15	50
C2	60	35	-	50	20	30	40	10
C3	35	30	45	-	20	50	60	35
C4	25	40	35	30	-	50	60	20
C5	30	35	10	60	35	-	80	20
C6	45	20	35	50	70	45	-	25
C7	60	35	25	40	30	60	50	-

Em que dispõe de veículos com 80 de capacidade e as procuras dos clientes são:

	C1	C2	C3	C4	C5	C6	C7
Procuras	45	50	30	20	25	35	30

a) Utilize o Solver para obter a solução do problema relaxado em que não se consideram as restrições de eliminação de subcircuitos ilegais.
b) Introduza restrições na resolução de a) para tentar melhorar o valor do minorante, caso possível.

CVRP – Modelos - CNO

- Dados: $G = (V, E)$; c_{ij}; q_i; K; Q;
- Variáveis: $x_{ij} = \begin{cases} 1 & \text{se 1 veículo viaja entre } i \in V \text{ e } j \in V \\ 0 & \text{c. c.} \end{cases}$

$\boxed{|E| = \frac{n(n+1)}{2} \text{ variáveis!}}$

G. Laporte, Y. Nobert, M. Desrochers 1985
Optimal routing under capacity and distance restrictions
Op. Res., Vol. 33, pp. 1050-1073

- Modelo:

$$(\text{VRP2}) \quad \min \sum_{\{i,j\} \in E} c_{ij}\, x_{ij}$$

Minimização do custo total

$$\text{s. a:} \begin{cases} \displaystyle\sum_{j \in \delta(i)} x_{ij} = 2 \quad \forall i \in N \\ \displaystyle\sum_{j \in \delta(0)} x_{0j} = 2|K| \\ \displaystyle\sum_{e \in \delta(S)} x_e \geq 2 \left\lceil \frac{q(S)}{Q} \right\rceil \quad \forall S \subseteq N, S \neq \emptyset \\ x_{ij} \in \{0,1\} \quad \forall \{i,j\} \in E \end{cases}$$

2 arestas incidentes em cada cliente i

Os $|K|$ veículos partem e chegam ao depósito

Restrições de eliminação de subcircuitos ilegais & capacidade

VRP – Modelos – CNO – exemplo 5

Considere a matriz de custos seguinte relativa a um problema de roteamento nos nodos

	Dep	C1	C2	C3	C4	C5	C6	C7
Dep	-	20	55	50	40	15	30	40
C1		-	40	20	50	25	15	50
C2			-	50	20	30	40	10
C3				-	20	50	60	35
C4					-	50	60	20
C5						-	80	20
C6							-	25
C7								-

Em que dispõe de veículos com 80 de capacidade e as procuras dos clientes são:

	C1	C2	C3	C4	C5	C6	C7
Procuras	45	50	30	20	25	35	30

a) Utilize o Solver para obter a solução do problema relaxado em que não se consideram as restrições de eliminação de subcircuitos ilegais.

b) Introduza restrições na resolução de a) para tentar melhorar o valor do minorante, caso possível.

CVRP – Modelos

- Técnicas para lidar com as restrições de quebra de subcircuitos em nº exponencial:
  - Relaxações – eliminando todas ou parte destas restrições, que podem depois ir sendo inseridas – Branch-and-Cut!
  - Recorrendo a novas variáveis & restrições – caso orientado – *MTZ-model*

D.L. Miller, A.W. Tucker, R.A. Zemlin, 1960
Integer programming formulations for the TSP
Journal of the ACM, Vol. 7, pp. 326-329

CVRP – Modelos - CO

- Restrições de eliminação de subcircuitos ilegais & capacidade:

$$\sum_{j \in \delta^+(S)} x_{ij} \geq \left\lceil \frac{q(S)}{Q} \right\rceil \quad \forall S \subseteq N, S \neq \emptyset$$

D.L. Miller, A.W. Tucker, R.A. Zemlin, 1960

- Novas Variáveis: u_i - procura já satisfeita quando o veículo chega a $i \in N$
  - Quebra de Subcircuitos ilegais: $u_i - u_j + Qx_{ij} \leq Q - q_j \quad \forall (i,j) \in A(N)$
  - Capacidade: $q_i \leq u_i \leq Q \qquad \forall i \in N$

CVRP – Modelos - CO

➢ Modelo: *MTZ*

$$(\mathrm{VRP}MTZ)\quad \min \sum_{(i,j)\in A} c_{ij}\, x_{ij}$$ Minimização do custo total

s. a:

$$\sum_{j\in\delta^+(i)} x_{ij} = 1 \quad \forall i \in N$$ Sai 1 vez de cada cliente i

$$\sum_{j\in\delta^-(i)} x_{ji} = 1 \quad \forall i \in N$$ Chega 1 vez a cada cliente i

$$\sum_{j\in\delta^+(0)} x_{0j} = |K|$$ São usados os $|K|$ veículos

$$u_i - u_j + Qx_{ij} \le Q - q_j \quad \forall (i,j) \in A(N)$$ Eliminação de subcircuitos ilegais

$$q_i \le u_i \le Q \quad \forall i \in N$$ Capacidade

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

$\boxed{n(n+2)\text{ variáveis}}$ $\boxed{n(n+5)+1\text{ restrições}}$

VRP – Modelos – CO – exemplo 4

Considere de novo o exemplo 4:

a) Formule o problema utilizando as restrições *MTZ* para eliminação dos subcircuitos ilegais

b) Compare os valores dos problemas resolvidos anteriormente com o de a)

LISBON SCHOOL OF ECONOMICS & MANAGEMENT
UNIVERSIDADE DE LISBOA

Cap. 2 **Problemas de Otimização Combinatória**
VRP – aula 8

SIMO/MQDEE

Maria Cândida Mourão

(cmourao@iseg.utl.pt)

CVRP – Modelos - CO

LISBON SCHOOL OF ECONOMICS & MANAGEMENT
UNIVERSIDADE DE LISBOA

Problema do modelos anteriores:

- Pode ser difícil identificar o serviço de cada veículo!

- Modelos com variáveis de 3 índices!

Bruce L. Golden
Univ. Maryland

Tom L. Magnanti
MIT

B.L. Golden, T.L. Magnanti, H.Q. Nguyen, 1977
Implementing VR algorithms
Netwotks, Vol. 7, pp. 113-148

CVRP – Modelos - CO

Modelos de 3 índices

B.L. Golden, T.L. Magnanti, H.Q. Nguyen, 1977

- Variáveis:
  - $x_{ij}^k = \begin{cases} 1 & \text{se o veículo } k \text{ viaja de } i \in V \text{ para } j \in V \\ 0 & \text{c. c.} \end{cases}$
  - $y_i^k = \begin{cases} 1 & \text{se o veículo } k \text{ visita } i \in V \\ 0 & \text{c. c.} \end{cases}$
  - u_i^k - (minorante da) procura servida pelo veículo k até chegar a $i \in N$

$|K|[n(n+3)+1]$ variáveis

CVRP – Modelos - CO

3 índices com restrições *MTZ*

$$(\text{VRP3}) \quad \min \sum_{k=1}^{|K|} \sum_{(i,j)\in A} c_{ij}\, x_{ij}^k$$

Minimização do custo total

s. a:

$$\sum_{j\in\delta^+(i)} x_{ij}^k - \sum_{j\in\delta^-(i)} x_{ji}^k = 0 \quad \forall i \in N; \forall k \in K$$

cada k, entra e sai de i = nº de vezes

$$\sum_{j\in\delta^+(0)} x_{0j}^k = 1 \quad \forall k \in K$$

cada k, sai 1 vez do depósito

$$\sum_{j\in\delta^+(i)} x_{ij}^k = y_i^k \quad \forall i \in N; \forall k \in K$$

cada k, se sai de i visita i

$$\sum_{j\in\delta^-(0)} x_{j0}^k = y_0^k \quad \forall k \in K$$

cada k, regressa ao depósito

$$\sum_{k\in K} \sum_{j\in\delta^+(i)} x_{ij}^k = 1 \quad \forall i \in N$$

ganrantir os serviços

$$u_i^k - u_j^k + Q x_{ij}^k \le Q - q_j \quad \forall (i,j) \in A; \forall k \in K$$

eliminação de subcircuitos ilegais

$$q_i \le u_i^k \le Q \quad \forall i \in N; \forall k \in K$$

capacidade

$$x_{ij}^k \in \{0,1\} \quad \forall (i,j) \in A; \forall k \in K$$

$$y_i^k \in \{0,1\} \quad \forall i \in N; \forall k \in K$$

$|K|[n(n+5)+2]+n$ restrições

VRP – Modelos – CO – exemplo 4

Considere de novo o exemplo 4:

Formule o problema utilizando as restrições *MTZ* para eliminação dos subcircuitos ilegais e a formulação de 3 índices, recorrendo a veículos com capacidade 150.

CVRP – Heurísticas Construtivas

- Greedy
- Cluster 1st – Route 2nd
  - Identificar uma partição do conjunto de clientes N em $|K|$ subconjuntos
  - Identificar uma rota admissível que sirva cada um dos subconjuntos de clientes - TSP
- Route 1st – Cluster 2nd
  - Identificar uma rota que inclua todos os clientes sem considerar as restrições de capacidade
  - Dividir a rota identificada em $|K|$ subrotas, uma para cada veículo

VRP – Savings Algorithm – Clarke & Wright

- H. de Savings - Greedy:

Clarke, G. & Wright, J.W., 1964
Scheduling of Vehicles from a Central Depot to a Number of Delivery Points
Op. Res., Vol. 12, pp. 568-581

Passo **0**: Inicialização:

Considerar que cada cliente é servido isolado, ou seja, numa rota que o inclui apenas a ele

Passo **1**: Calcular o *saving* de juntar cada par de clientes por: $s_{ij} = c_{i0} + c_{0j} - c_{ij}$

Passo **2**: Ordenar os *savings* por ordem decrescente numa lista L

Passo **3**: Selecionar as rotas associada à junção de maior *saving* (1ª da lista L)

Se a junção das rotas identificadas for possível (Q),

- juntar as rotas respetivas
- Atualizar L; recalculando os *savings* necessários
- Voltar a 2

C.c., retirar a junção de L

Passo **4**: **Se** ainda há *savings* positivos em L, voltar a 2

c.c. FIM

VRP – Modelos – CNO – exemplo 5

Considere de novo o exemplo 5.

a) Utilizando a heurística de *savings* identifique uma SA.

b) Compare os valores dos problemas resolvidos anteriormente com o de a)

VRP –Extensões

- ✓ Time Windows – Clientes e Depósito
- ✓ Frota Heterogénea
- ✓ Periódico
- ✓ Pickup & Delivery – restrições de carregamento
- ✓ Com Backhauls
- ✓ Problemas Estocásticos
- ✓ Green VRP
- ✓ Problemas Dinâmicos: procuras dinâmicas; tempos/procuras dependentes do tempo
- ✓ VRP Solver

VRP – Trabalho

Proponha um enunciado para um problema de VRP, recorrendo a uma instância com 7 clientes e 2 veículos.

a) Utilize um algoritmo para obter uma SA.

b) Proponha, recorrendo a uma relaxação, um minorante para o problema.

c) Introduza um corte na relaxação proposta para tentar melhorar o valor do minorante.

d) Utilizando o modelo de três índices e as restrições *MTZ*, para impedir subcircuitos ilegais, resolva o problema.

e) Utilize o *VRP Solver* para obter uma solução e compare-a com as geradas nas alíneas anteriores.

Cap. 2 **Problemas de Otimização Combinatória**
VRP – aula 9

SIMO/MQDEE

Maria Cândida Mourão

(cmourao@iseg.utl.pt)

VRPTW –CO

TIME WINDOWS

- Dados:
  - Cada cliente $i \in N$ deve ser visitado no intervalo de tempo $[a_i, b_i]$
  - Depósito representado pelos nodos 0 e $n+1$
  - s_i tempo necessário para servir o cliente $i \in N$ $(s_0 = s_{n+1} = 0)$
  - t_{ij} tempo de viagem de $i \in V$ para $j \in V$

- Variáveis:
  - $x_{ij}^k = \begin{cases} 1 & \text{se o veículo } k \text{ viaja de } i \in V \text{ para } j \in V \\ 0 & \text{c. c.} \end{cases}$
  - T_i^k - instante de início de serviço a $i \in N$ pelo veículo $k \in K$

$\boxed{|K|n(n+2) \text{ variáveis}}$

VRPTW – Modelos - CO

(VRPTW) $\min \sum_{k\in K} \sum_{(i,j)\in A} c_{ij}\, x_{ij}^k$ Minimização do custo total

s. a:

$\sum_{j\in\delta^+(i)} x_{ij}^k - \sum_{j\in\delta^-(i)} x_{ji}^k = 0 \quad \forall i \in N; \forall k \in K$ cada k, entra e sai de i = nº de vezes

$\sum_{j\in\delta^+(0)} x_{0j}^k = 1 \quad \forall k \in K$ cada k, sai 1 vez do depósito

$\sum_{k\in K} \sum_{j\in\delta^+(i)} x_{ij}^k = 1 \quad \forall i \in N$ cada i é visitado por um só k

$\sum_{j\in\delta^-(n+1)} x_{j,n+1}^k = 1 \quad \forall k \in K$ cada k, regressa ao depósito

$\sum_{i\in N} q_i \sum_{j\in\delta^+(i)} x_{ij}^k \le Q \quad \forall k \in K$ capacidade

$T_i^k + s_i + t_{ij} - T_j^k + M_{ij} x_{ij}^k \le M_{ij} \quad \forall (i,j) \in A; \forall k \in K$ TW & eliminação de subcircuitos

$a_i \sum_{j\in\delta^+(i)} x_{ij}^k \le T_i^k \le b_i \sum_{j\in\delta^+(i)} x_{ij}^k \quad \forall i \in N; \forall k \in K$ TW

$x_{ij}^k \in \{0,1\} \quad \forall (i,j) \in A; \forall k \in K$

$\boxed{|K|(2n^2 + 3n + 3) \text{ restrições}}$

VRPTW – CO – exemplo 6

Considere a matriz de tempos seguinte relativa a um problema de roteamento nos nodos

	Dep	C1	C2	C3	C4	C5	C6	C7
Dep	-	10	8	7	6	4	5	6
C1	8	-	4	2	5	3	2	5
C2	10	4	-	5	2	3	4	1
C3	6	3	4	-	2	5	6	4
C4	5	4	3	3	-	5	6	2
C5	6	3	1	6	4	-	8	2
C6	7	2	5	5	7	5	-	3
C7	8	5	2	4	3	6	5	-

Em que dispõe de veículos com 130 de capacidade e as TW e as procuras dos clientes são:

	C1	C2	C3	C4	C5	C6	C7
Procuras	45	50	30	20	25	35	30
TW	[2, 15]	[8, 20]	[10, 30]	[2, 15]	[10, 30]	[10, 30]	[15, 40]
Tempos serviço	2	4	2	3	4	2	3

Utilize o Solver para resolver o problema com o modelo apropriado.

VRP – SOLVER

- Excel Add-in
- VRP_Spreadsheet_Solver_v3.01 – Exemplo Lisboa!

G Erdoğan, 2017

An open source Spreadsheet Solver for VRPs

Comp. & Operations Research, Vol. 84, pp. 62-72

Problemas de Roteamento

Cap. 2 – Problemas de Otimização Combinatória - Roteamento

Problemas de roteamento nos nodos

Problemas de roteamento nos arcos

Utilização de Software – VRP **Spreadsheet Solver**

Gilbert Laporte
HEC Montréal

Ángel Corberán
Univ. Valencia

- ARP
  - Relaxações
  - Heurísticas

Á. Corberán & G. Laporte (2014)
Arc Routing Problems, Methods, and Applications
MOS-SIAM Series on Optimization, Philadelphia

J.R. Evans & E. Minieka (1992)
Optimization Algorithms for Networks and Graphs
2bd edition, Marcel Dekker, Inc. NY

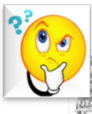
Roteamento nos Arcos – ARP

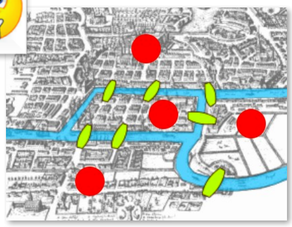
História

- 1º problema – 7 pontes de Königsberg (Kaliningrad)
- Ilha de *Kneiphof* – 2 braços de rio a rodear a ilha – 7 pontes!

Leonard Euler
1707-1783

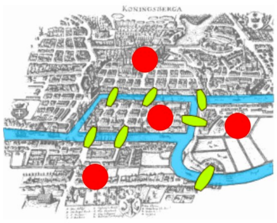
Pergunta – é possível identificar um caminho que passe uma e uma só vez por cada ponte?



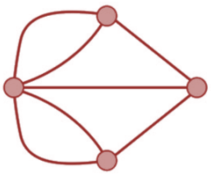


Á. Corberán & G. Laporte (2014)
A Historical Perspective on Arc Routing
in
Arc Routing book, same autors

Roteamento nos Arcos – ARP

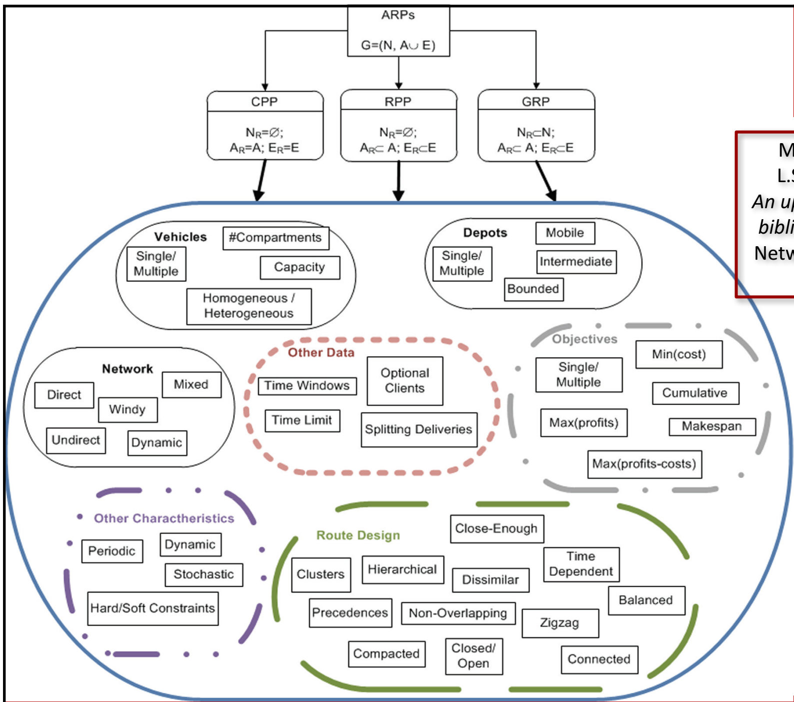


Grafo



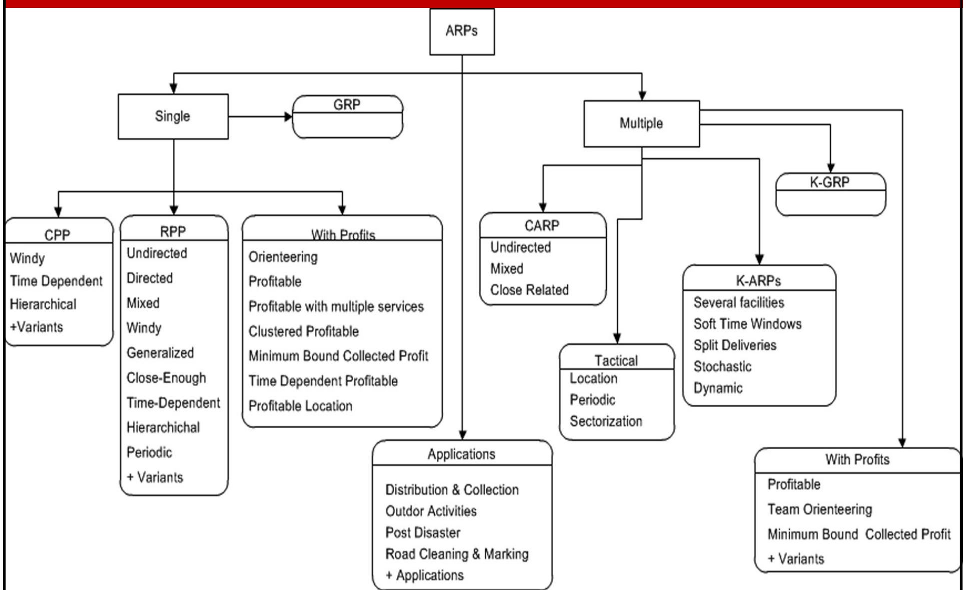
7 pontes de Königsberg – Resposta:

- Se existirem + de 2 áreas (2 nodos) com um nº ímpar de pontes (ligações) – Impossível
- Se existirem exatamente 2 áreas com um nº ímpar de pontes – possível se o caminho começar numa dessas áreas e terminar na outra
- Se todas as áreas estiverem ligadas a um nº par de pontes – possível com início e fim em qualquer das áreas



M.C. Mourão & L.S. Pinto (2017)
An updated annotated bibliography on ARPs
Networks, Volume 70, pp. 144-194

Roteamento nos Arcos – ARP



Roteamento nos Arcos – CPP

Circuito (Ciclo) Euleriano (CE) – Passa uma e uma só vez por cada ligação do grafo

Objetivos:

- Transformar o grafo num grafo Euleriano
- Identificar no grafo Euleriano um CE

Roteamento nos Arcos – CPP

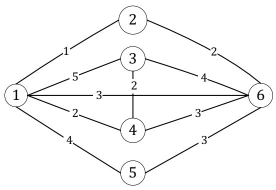
Resolução – CNO:

1. Identificar um *matching* perfeito de custo mínimo entre todos os vértices de grau ímpar do grafo (são em nº par!)
2. Juntar as arestas do matching ao grafo inicial

Carteiro Chinês

Exemplo

Resolva o problema do CPP considerando a rede, onde os valores sobre as ligações representam as respetivas distâncias, e a respetiva matriz de caminhos mais curtos entre qualquer par de vértices.



	1	2	3	4	5	6
1	0	1	4	2	4	3
2	1	0	5	3	5	2
3	4	5	0	2	7	4
4	2	3	2	0	6	3
5	4	5	7	6	0	3
6	3	2	4	3	3	0

Roteamento nos Arcos – CPP

Grafo Simétrico – Grau interno de cada vértice iguala o seu grau externo, ou seja: $\delta^-(i) = \delta^+(i), \quad \forall i \in V$

Objetivos:

- Transformar o grafo num grafo Euleriano - simétrico
- Identificar no grafo Euleriano um CE

Roteamento nos Arcos – CPP

L. Bodin
Univ. Maryland

Resolução – CO:

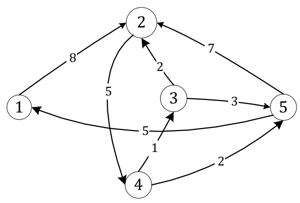
1. Resolver um problema de transporte (PT), considerando:
   i. Origens – vértices com mais arcos a entrar que a sair, com oferta: $\delta^{-}(i) - \delta^{+}(i)$
   ii. Destinos – vértices com mais arcos a sair que a entrar, com procura: $\delta^{+}(i) - \delta^{-}(i)$
   iii. Custo de cada ligação origem, destino são os custos do caminho de menor custo dessa origem para esse destino
2. Juntar os arcos associados à SO do PT (cada arco representa um caminho mais curto entre os respetivos vértices origem, destino) resolvido em 1. ao grafo inicial

E. Beltrami & L. Bodin (1974)
Networks and VR for Municipal Waste Collection
Networks, Volume 4, pp. 65-94

Carteiro Chinês

Exemplo

Resolva o problema do CPP considerando a rede, onde os valores sobre as ligações representam as respetivas distâncias, e a respetiva matriz de caminhos mais curtos entre qualquer par de vértices.



	1	2	3	4	5
1	0	8	14	13	15
2	12	0	6	5	7
3	8	2	0	7	3
4	7	3	1	0	2
5	5	7	13	12	0

Cap. 2 Problemas de Otimização Combinatória

SIMO/MQDEE

F I M

Cap. 2 Problemas de Otimização Combinatória

Problema(s) a estudar

Aplicações

Formalização

Heurística – falar de várias e apresentar uma!

Exercício(s)

Software – Open Solver – fazer sempre um exemplo no software –

Heurísticas construtivas simples: savings; vizinho + perto, vizinho + longe, melhorativa; route 1st/cluster 2nd vs cluster 1st/route second

1 veículo *vs* k veículos

Distinção de redes: orientada; não orientada; mista

Cada grupo apresenta um problema prático e a resolução no vrp solver – Todos menos este grupo resolvem o caso prático que eu der e entregam a respetiva resolução.

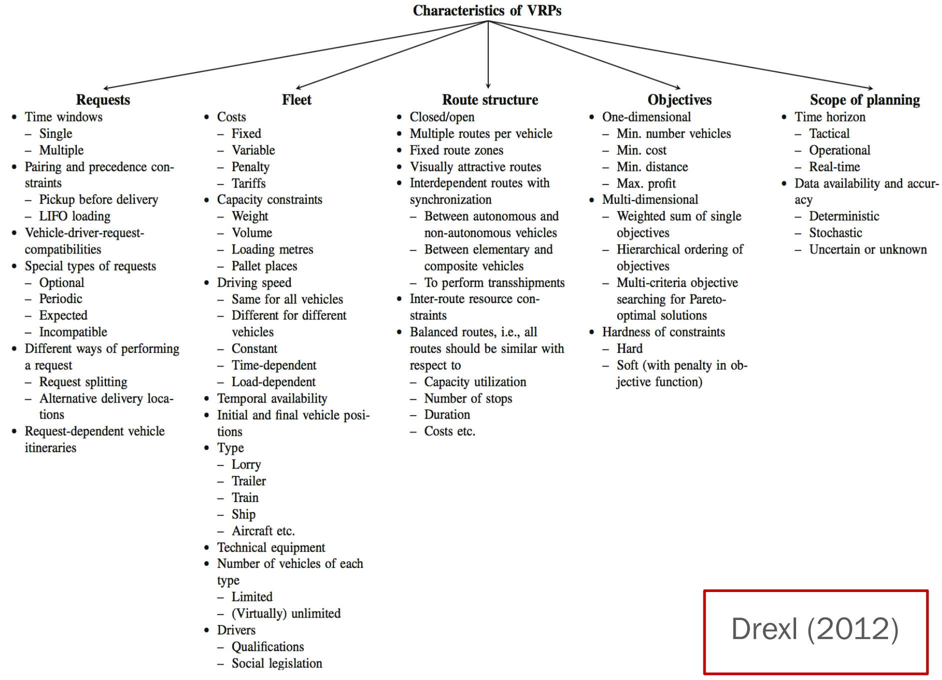


Fig. 1 Dimensions of richness in VRPs

VRP - Software

Free software for solving VRP[edit]

Name (alphabetically)	License	API language	Brief info
jsprit	Apache License	Java	lightweight, java based, open source toolkit for solving rich VRPs. link An Excel-compatible user interface for jsprit is available with mapping, reporting and route editing functionality. link [10]
Open-VRP	LGPL	Lisp	Open-VRP for Lisp, hosted on Github. link [10]
OptaPlanner	Apache License	Java	Open Source Java constraint solver (optaplanner.org) with VRP examples. [10]
SYMPHONY	Common Public License 1.0		Open-source solver for mixed-integer linear programs (MILPs) with support for VRPs. link [10]
FAST			Interactive vehicle routing software for small vehicle fleets intended for use by small businesses: FAST
VRP Spreadsheet Solver	Creative Commons Attribution 4.0 International License		Microsoft Excel and VBA based open source solver, with a link to public GIS for coordinate, driving distance and duration retrieval. link Video tutorial link [10]

TSP – Modelos – CO - exemplo 3

Considere a matriz de custos seguinte relativa a um problema de Caixeiro viajante

$$\begin{array}{c|c} i,j & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} \infty & 10 & 3 & 6 & 9 \\ 5 & \infty & 5 & 4 & 2 \\ 4 & 9 & \infty & 7 & 8 \\ 7 & 1 & 3 & \infty & 4 \\ 3 & 2 & 6 & 5 & \infty \end{pmatrix} \end{array}$$

	C1	C2	C3	C4	C5
C1	∞	3	7	6	2
C2	4	∞	11	9	6
C3	2	6	∞	1	5
C4	5	5	3	∞	4
C5	6	2	7	6	∞

a) Formalizar o problema

b) Utilize o Solver para obter a solução do problema relaxado em que não se consideram as restrições de eliminação de subcircuitos

c) Introduzir restrições de eliminação de subcircuitos até encontrar uma SO do TSP