

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:

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

$$\text{s. a: } \begin{cases} \sum_{j:(i,j) \in A} x_{ij} = 1 \quad \forall i \in V & \text{De cada nodo } i \text{ só sai uma ligação} \\ \sum_{k:(k,i) \in A} x_{ki} = 1 \quad \forall i \in V & \text{Só uma ligação é usada para entrar em cada nodo } i \\ \sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1 \quad \forall S \subset V & \text{Restrições de eliminação de subcircuitos ilegais} \\ x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \end{cases}$$

TSP – Modelos – CO – exemplo 3

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

$$\begin{array}{c}
 i, j \quad \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\
 \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \quad \left(\begin{array}{cccc}
 \infty & 3 & 5 & 1 \\
 \infty & \infty & 1 & 6 \\
 6 & 1 & \infty & 3 \\
 1 & 5 & \infty & \infty
 \end{array} \right)
 \end{array}$$

- 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:

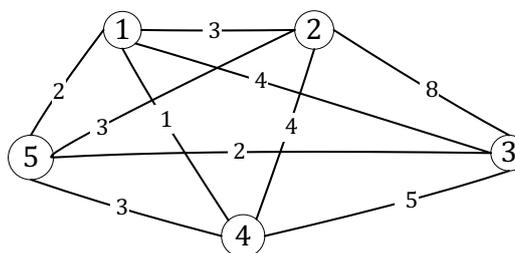
$$(P2) \min \sum_{e \in E} c_e x_e \quad \text{Minimização do custo total}$$

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

TSP – Modelos – CNO - exemplo 1

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

i, j	1	2	3	4	5
1	–	3	4	1	2
2		–	8	4	3
3			–	5	2
4				–	3
5					–



- Formalize o problema
- 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.
- Compare os valores da solução obtida com o majorante obtido pela H. de Inserção mais próxima.



TSP –Relaxações – caso 0 (assimétrico)

- $\forall SA$ verifica:
 - Cada nodo tem um arco a “entrar” e um a “sair”!
 - O circuito é conexo
- Relaxação:

(P1R1) $\min \sum_{(i,j) \in A} c_{ij} x_{ij}$

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

~~Restrições de eliminação de subcircuitos ilegais~~

Minimização do custo total

De cada nodo i só sai uma ligação

Só uma ligação é usada para entrar em cada nodo i

Problema de Afetação (PA)

SIMULAÇÃO E OTIMIZAÇÃO (MODEE) – 2017/18
46



TSP –Relaxações – caso 0 (assimétrico)

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!

SIMULAÇÃO E OTIMIZAÇÃO (MODEE) – 2017/18
47

TSP – Relaxações – exemplos

- Defina o PA que representa uma relaxação para o exemplo 3 (CO)
- Resolva o problema de a) pelo Solver/Excel e, se necessário, introduzir cortes (quebras de subcircuitos ilegais) para encontrar a SO do problema inicial.
- Defina o PA que representa uma relaxação para o exemplo 1 (CNO)
- 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

- \forall 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

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

$$\text{s. a: } \begin{cases} \sum_{j:(j,i) \in A} x_{ji} = 1 \quad \forall i \in V & \text{Só uma ligação é usada para entrar em cada nodo } i \\ \sum_{i \in S} \sum_{j \in V \setminus S} x_{ij} \geq 1 \quad \forall S \subset V & \text{Restrições de eliminação de subcircuitos ilegais} \\ x_{ij} \in \{0,1\} \quad \forall (i,j) \in A \end{cases}$$

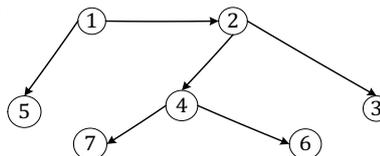
\forall SA - 1 arco a “entrar” em cada nodo, num grafo conexo

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 raiz 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

- Defina, a partir de uma arborescência geradora mínima, uma relaxação para o exemplo 3 e determine o correspondente minorante.
- Compare o valor de a) com o minorante calculado pela resolução do PA.
- 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

- $\forall SA$ 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

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

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

$\forall SA - n$ arestas que formam um grafo conexo

TSP –Relaxações – CNO



Algoritmo – Relaxação:

- Remover um vértice qualquer x de G
- Identificar uma árvore geradora no grafo 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 árvore-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**

SIMULAÇÃO E OTIMIZAÇÃO (MODEE) – 2017/18
56



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.

SIMULAÇÃO E OTIMIZAÇÃO (MODEE) – 2017/18

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	∞

- Formalize o problema
- Identifique uma SA pelo método de Inserção Mais Próxima
- Identifique um minorante recorrendo à resolução de um PA.
- Introduza, na solução de c) restrições relaxadas até encontrar uma SO do TSP.
- Repita b) e c) considerando a relaxação que resulta na resolução de um problema de arborescência.