Ata de resolução do exercício t3.2.9 (*Hungarian Algorithm*)

Nome: Gustavo Waku
Data: 13-abr-2012

O algoritmo húngaro busca encontrar um matching perfeito de peso máximo (transversal máxima). A seguir veremos 3 abordagens de resolução desse problema.

Método Manual com planilha

Entradas:

a)

4	4	4	3	6
1	1	4	3	4
1	4	5	3	5
5	6	4	7	9
5	3	6	8	3

b)

7	8	9	8	7
8	7	6	7	6
9	6	5	4	6
8	5	7	6	4
7	6	5	5	5

c)

1	2	3	4	5
6	7	8	7	2
1	3	4	4	5
3	6	2	8	7
4	1	3	5	4

Durante a aula, com o auxílio de uma planilha (em anexo) nós elaboramos uma matriz que basicamente faz o cálculo da matriz de excessos.

Fizemos manipulações na linha e na coluna, tentando colocar zeros nas linhas e colunas, de forma que encontrássemos uma cobertura mínima de vértices (que contenha todos zeros da matriz) de tamanho igual ao número de vértices.

A somatória das coberturas (u e v) dessa matriz de excessos nos mostra o valor da soma máxima da matriz original (no canto superior esquerdo da matriz). Cada zero na matriz de excessos representa um possível elemento que irá compor a transversal máxima. Observando que a escolha de um zero, invalida a escolha de um outro zero na mesma linha ou coluna.

Os zeros sublinhados são as posições dos valores da matriz original de pesos que compõe a transversal máxima.

Na matriz a, encontramos o seguinte:

29	2	2	3	4	5
2	<u>0</u>	0	1	3	1
1	2	2	<u>0</u>	2	2
2	3	<u>0</u>	0	3	2
4	1	0	3	1	<u>0</u>
4	1	3	1	<u>0</u>	6

29 é a soma da dos índices u e v da matriz de excessos e o valor que procuramos.
Portanto, a transversal máxima = 4 + 4 + 4 + 9 + 8 = 29.

Na matriz b, temos:

36	6	4	5	4	3
4	3	0	<u>0</u>	0	0
3	1	<u>0</u>	2	0	0
3	<u>0</u>	1	3	3	0
2	0	1	0	<u>0</u>	1
2	1	0	2	1	<u>0</u>

Transversal máxima: 9 + 7 + 9 +6 + 5 = 36.

Na matriz c, temos:

28	0	1	2	3	4
1	0	0	0	0	<u>0</u>
6	0	0	<u>0</u>	2	8
2	1	<u>0</u>	0	1	1
5	2	0	5	<u>0</u>	2
4	<u>0</u>	4	3	2	4

Transversal máxima: 5 + 8 + 3 + 8 + 4 = 28.

Método Manual sem planilha

Para abordagem de um método manual, vamos utilizar o pseudo-código do Algoritmo Húngaro baseado nas instruções do exemplo 3.2.10 do nosso livro. A idéia é bem parecida com a abordagem de planilha, mas vamos executar os ajustes da cobertura de uma forma mais estruturada.

Algoritmo Húngaro (matriz_pesos w)

Matriz de excessos c
Cobertura u, Cobertura v
$u_i \leftarrow max_j (w_{i,j})$ //iniciar cobertura i com o maior valor da linha na matriz de pesos
$v_j \leftarrow 0$ //iniciar a cobertura v com zero em todos os elementos.
$c_{i,j} \leftarrow u_i + v_j - w_{i,j}$ //calcula matriz de excessos
analisar a matriz de excessos c e achar uma cobertura de vértices Q. Q deve ser mínimo e deve conter todos os zeros da matriz de excessos.

while ($|Q| < n$) // n de $K_{n,n}$ da matriz w
$R \leftarrow X \cap Q$ //vertices que estão na cobertura Q em X
$T \leftarrow Y \cap Q$ //vertices que estão na cobertura Q em Y

incremento $\leftarrow \min \{ u_i + v_j - w_{i,j} : x_i \in X - R ,\ y_j \in Y - T \}$ /*pega o menor valor da matriz de pesos formada a partir da exclusão das colunas que pertencem à cobertura em Y e das linhas que pertencem à cobertura em X. */

//atualiza Coberturas u e v
for each $x_i \in X - R$ **do**
$u_i \leftarrow u_i$ – incremento

for each $y_j \in T$ **do**
$v_j \leftarrow v_j$ + incremento

$c_{i,j} \leftarrow u_i + v_j - w_{i,j}$ /* atualiza a matriz de excessos a partir dos ajustes nas coberturas analisar a matriz de excessos c e achar nova cobertura Q; */

achar uma transversal formada por zeros na matriz c. /* Os índices (i e j) dessa transversal na matriz c correspondem ao valor de cada transversal na matriz de pesos original. */

Exemplo passo a passo do exercício **3.2.5.c**, executando o algoritmo acima.

Matriz de pesos original:

1	2	3	4	5
6	7	8	7	2
1	3	4	4	5
3	6	2	8	7
4	1	3	5	4

1) Matriz de excessos e as respectivas coberturas iniciais:

	0	0	0	0	0		
5	4	3	2	1	0		
8	2	1	0	1	6	R	<--
5	4	2	1	1	0		
8	5	2	6	0	1		
5	1	4	2	0	1		
				T	T		
				^	^		

O valor em laranja indica o valor do incremento mínimo, excluindo as linhas R e colunas T. As setas indicam a cobertura Q.

2) recalculando a matriz de excessos a achando nova cobertura:

	0	0	0	1	1		
4	3	2	1	1	0		
8	2	1	0	2	7		
4	3	1	0	1	0		
7	4	1	5	0	1		
4	0	3	1	0	1	R	<--
			T	T	T		
			^	^	^		

3) recalculando novamente a matriz de excessos e a cobertura:

	0	0	1	2	2
3	2	1	1	1	$\underline{0}$
7	1	$\underline{0}$	0	2	7
3	2	0	$\underline{0}$	1	0
6	3	0	5	$\underline{0}$	1
4	$\underline{0}$	3	2	1	2
	T	T	T	T	T
	^	^	^	^	^

Temos que |Q| = 5. O algoritmo termina o laço principal. Os valores sublinhados são os valores quem compõem a transversal máxima: 5 + 7 + 4 + 8 + 4 = 28.

Abordagem via programa

O aluno Marlon Fernandes desenvolveu um programa (hungarian.cpp em anexo) que implementa o algoritmo húngaro.

O programa segue os princípios do algoritmo 3.2.9. A função principal é getMaxCost() que retorna o custo máximo; em init_labels(), inicializa as coberturas lx com o valor máximo de cada linha da matriz de pesos e ly com zeros (análogo às coberturas u e v); depois chama a função augment(). Esta função acha um caminho aumentante incrementalmente, atualiza os excessos baseado em R e T; e termina quando o matching é máximo.

As saídas do programa:

Para a matriz a:

"RESULTADO = 29

MATCHING:
1 -> 1
2 -> 3
3 -> 2
4 -> 5
5 -> 4"

Matriz b:
"RESULTADO = 36

MATCHING:
1 -> 3
2 -> 2
3 -> 1
4 -> 4
5 -> 5"

Matriz c:
"RESULTADO = 28

MATCHING:
1 -> 5
2 -> 3
3 -> 2
4 -> 4
5 -> 1"

Interessante notar que as saídas do programa foram exatamente as mesmas que encontramos na abordagem com a planilha.