10/7/13

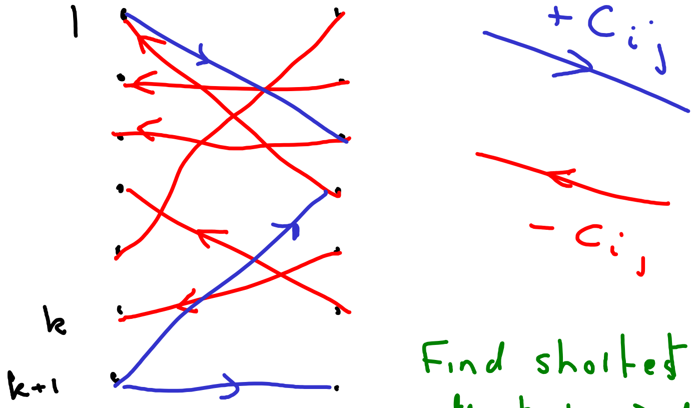
Assignment problem

Repeat: $k = 1, 2, \ldots, n$

Solve $k \times k$ problem

$\downarrow$

Set shortest path problem for $k+1 \times k+1$



Find shortest path $k+1 \longrightarrow k+1$

Making arc lengths of S.P. problem non-negative.

Replace c_{ij} by $\hat{c}_{ij} = c_{ij} + \lambda_i - \lambda_j$

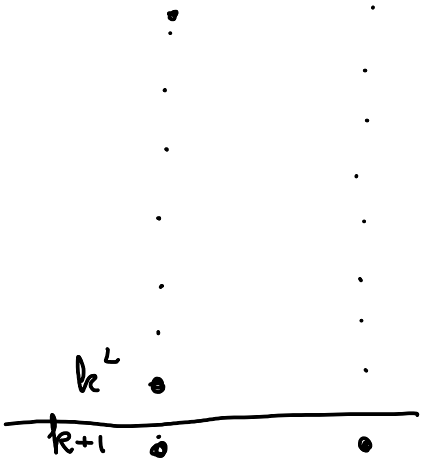
So, if I can find λ's s.t. $\hat{c}_{ij} \geq 0$ $\forall i, j$.. then I can use Dijkstra algorithm

Suppose P is a path from x to y

$$\hat{\ell}(P) = c_{x_1, x_2} + \lambda_{x_1} - \lambda_{x_2} + c_{x_2, x_3} + \lambda_{x_2} - \lambda_{x_3} + \cdots$$

$P = (x = x_1, x_2, \ldots, x_m = y)$

$$= \ell(P) + \lambda_x - \lambda_y$$



λ_i = length of shortest path from k^L to i then

$$\lambda_i + c_{ij} \geq \lambda_j$$

Optimality for shortest paths

(i) Need to take care of λ^L_{k+1} & λ^R_{k+1}

(ii) Need to deal with edges in matching - equality

Assignment problem as an integer program:

$$x_{ij} = \begin{cases} 0 : & \text{otherwise} \\ 1 : & i \text{ on left assigned to } j \text{ on right} \end{cases}$$

Minimise $\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$

s.t.

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \qquad 0 \leq x_{ij} \leq 1$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \forall j \qquad \boxed{x_{ij} \text{ integer}}$$

Always true for a basic solution

Matrix of constraints

$$A = \left[\begin{array}{ccccccccc} 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \\ \hline 1 & & & 1 & & & 1 & & \\ & 1 & & & 1 & & & 1 & \\ & & 1 & & & 1 & & & 1 \end{array}\right]$$

$\underline{n=3}$

Totally unimodular matrix: if B is a $k \times k$ submatrix of A then $\det|B| \in \{0, +1, -1\}$

Suppose we have a matrix with

(i) Only has $0, \pm 1$ entries

(ii) At most 2 non-zero entries per column

(iii) Partition of rows into 2 sets R_1, R_2 s.t. if column has $+1$ in R_1 then can only have $+1$ in R_2.

(iv) if column has a $+1$ & a -1 then one is in R_1 & other in R_2

$$
\begin{array}{c}
R_1 \\ \\ \\ R_2 \\ \\
\end{array}
\left[\begin{array}{cccc}
1 & & & 1 \\
 & 1 & & \\
 & -1 & & \\
\hline
 & & 1 & \\
1 & & -1 & \\
\end{array}\right]
$$

B is a $k \times k$ sub-matrix

$k=1$: $\det(B) = 0, \pm 1$

Induction

$\left[\quad B \quad\right]$ — every column has 2 non-zero's

$$
\left[\begin{array}{ccc}
 & 1 & \\
1 & -1 & 1 \\
\hline
1 & & 1
\end{array}\right]
$$

$\det(B) = 0$, since sum of R_1 rows = sum of R_2 rows.

B has a column with a single non-zero: $\begin{bmatrix} 1 & \\ & B' \end{bmatrix}$, $\det B = \det B' = 0, \pm 1$

Basic solution of a linear program:

Constraints are $A\underline{x} = \underline{b}$

$$A = [B : N]$$

$\underline{x}_B$ $\underline{x}_N$

Basic variables

$$B\underline{x}_B + N\underline{x}_N = \underline{b}$$

$\underline{x}_N = 0$

$$\underline{x}_B = B^{-1}\underline{b}$$

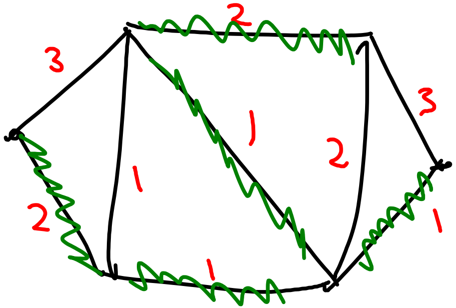
$$B^{-1} = \frac{\mathrm{adj}(B)}{\det B} \quad \begin{matrix} \leftarrow \text{integer matrix} \\ \leftarrow \pm 1 \end{matrix} \quad \Rightarrow B^{-1} \text{ is integer.}$$

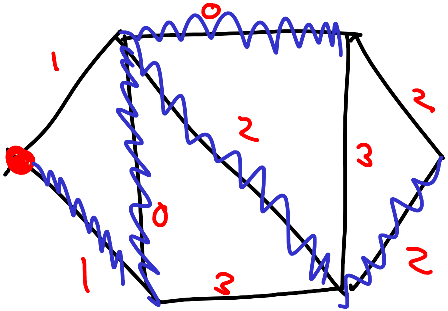
If A is totally unimodular then Integer Programming is no harder than linear programming.

Minimum Spanning Tree problem

Given a connected graph $G = (V, E)$ and $w : E \to \mathbb{R}$

find a minimum weight spanning tree.





Dijkstra: take cheapest edge leaving current tree.

Algorithm

Suppose we have selected $e_1, e_2, \dots, e_k$

Suppose graph induced by $e_1, e_2, \dots e_k$ has components $C_1, C_2, \dots C_l$

Choose a component C. Then choose cheapest edge leaving C.