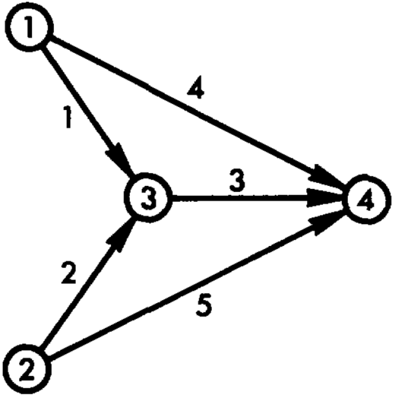
Multiple origin-destination pairs

Best explained by a simple example — in generality, the indexing notation becomes quite unpleasant!!!



- Inspired by Sheffi p79
- No 'unmixing' (a simple case)

- Link cost functions:

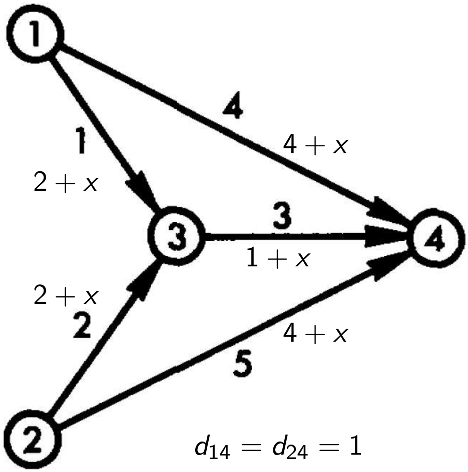
$c_1 = 2 + x_1$
$c_2 = 2 + x_2$
$c_3 = 1 + x_3$
$c_4 = 4 + x_4$
$c_5 = 4 + x_5$

- OD demands:

$d_{14} = 1$
$d_{24} = 1$ (say)

- Notice the top-down symmetry — to help hand calculations. But point is to formulate general problem (e.g., break symmetry in demand?) and give to Matlab.

Example — preliminaries



- Find (here, all) routes (here, in links):

route 1: $\{4\}$
route 2: $\{1, 3\}$
route 3: $\{5\}$
route 4: $\{2, 3\}$

- Route variables $\mathbf{y} \geq \mathbf{0}$ satisfy demands:

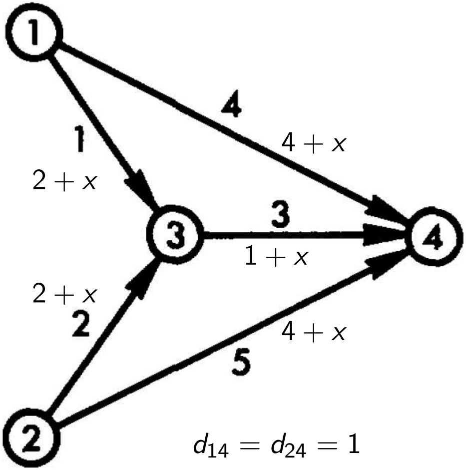
$$y_1 + y_2 = d_{14} \quad \text{and} \quad y_3 + y_4 = d_{24}.$$

- Find link-route incidence matrix:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(NB unusual — more links than routes.)

UE solution — complementarity method



R1: $\{4\}$. R2: $\{1, 3\}$. R3: $\{5\}$. R4: $\{2, 3\}$

- Compute link costs in route variables — formally $\mathbf{c}(\mathbf{Ay})$ — then compute route costs in route variables, is $\mathbf{A}^{\mathrm{T}}\mathbf{c}(\mathbf{Ay})$:
  R1: $4 + y_1$
  R2: $(2 + y_2) + (1 + y_2 + y_4)$
  $\quad = 3 + 2y_2 + y_4$
  R3: $4 + y_3$
  R4: $3 + y_2 + 2y_4$
- (Complementarity) Solve:
  $4 + y_1 = 3 + 2y_2 + y_4$
  $4 + y_3 = 3 + y_2 + 2y_4$
  with $y_1 + y_2 = d_{14} = 1$ and
  $y_3 + y_4 = d_{24} = 1$ and $\mathbf{y} \geq \mathbf{0}$.
- Solution $y_1 = y_2 = y_3 = y_4 = 1/2$.
  (Symmetry helps a lot!!!)
  Usually — need to check for unused routes.

UE solution — in route variables, by Beckmann form

- Beckmann function:

$$\hat{f}(\mathbf{x}) = \sum_{\text{links } i} \int_0^{x_i} c_i(\tilde{x})\,\mathrm{d}\tilde{x},$$

$$= \left(\frac{1}{2}x_1^2 + 2x_1\right) + \left(\frac{1}{2}x_2^2 + 2x_2\right) + \left(\frac{1}{2}x_3^2 + x_3\right) + \left(\frac{1}{2}x_4^2 + 4x_4\right) + \left(\frac{1}{2}x_5^2 + 4x_5\right),$$

$$= \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x} + \mathbf{a}^{\mathrm{T}}\mathbf{x},$$

where $\mathbf{H} = \mathbf{I} = \mathrm{diag}(\mathbf{b})$ and $\mathbf{a} = (2, 2, 1, 4, 4)^{\mathrm{T}}$.

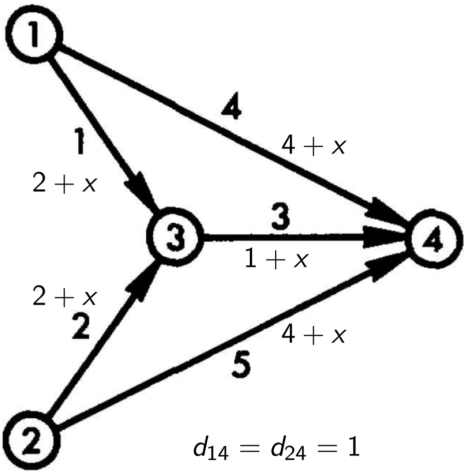
- Write $\mathbf{x} = \mathbf{A}\mathbf{y}$, and minimise subject to $\mathbf{y} \geq \mathbf{0}$ and demand constraints $y_1 + y_2 = d_{14}$ and $y_3 + y_4 = d_{24}$, or in vector form

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \mathbf{y} = \begin{pmatrix} d_{14} \\ d_{24} \end{pmatrix},$$

that is $\mathbf{M}\mathbf{y} = \mathbf{d}$, where $\mathbf{M}$ is **OD-route incidence matrix**.

- SO solution approach is similar: but $\mathbf{H} = 2\,\mathrm{diag}(\mathbf{b})$ (multiply instead of integrate).

UE solution — in Beckmann form, link variables only



$d_{14} = d_{24} = 1$

- All the route computations are skipped: saves a lot of work!!!

- Minimise

$$\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x} + \mathbf{a}^{\mathrm{T}}\mathbf{x}$$

subject to $\mathbf{x} \geq \mathbf{0}$ and node balances $\mathbf{B}\mathbf{x} = \mathbf{s}$, in the form

$$\begin{pmatrix} -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 \\ +1 & +1 & -1 & 0 & 0 \\ 0 & 0 & +1 & +1 & +1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} -d_{14} \\ -d_{24} \\ 0 \\ d_{14} + d_{24} \end{pmatrix}$$

- The only catch is for examples with flow 'unmixing' — not needed here.

With a little help from Matlab . . .

```
>> help quadprog
 quadprog Quadratic programming.
    X = quadprog(H,f,A,b) attempts to solve the quadratic programming
    problem:

             min 0.5*x'*H*x + f'*x   subject to:  A*x <= b
              x

    X = quadprog(H,f,A,b,Aeq,beq) solves the problem above while
    additionally satisfying the equality constraints Aeq*x = beq. (Set A=[]
    and B=[] if no inequalities exist.)

    X = quadprog(H,f,A,b,Aeq,beq,LB,UB) defines a set of lower and upper
    bounds on the design variables, X, so that the solution is in the
    range LB <= X <= UB. Use empty matrices for LB and UB if no bounds
    exist. Set LB(i) = -Inf if X(i) is unbounded below; set UB(i) = Inf if
    X(i) is unbounded above.

    X = quadprog(H,f,A,b,Aeq,beq,LB,UB,X0) sets the starting point to X0.

    X = quadprog(H,f,A,b,Aeq,beq,LB,UB,X0,OPTIONS) minimizes with the
    default optimization parameters replaced by values in OPTIONS, an
    argument created with the OPTIMOPTIONS function. See OPTIMOPTIONS for
    details.
```