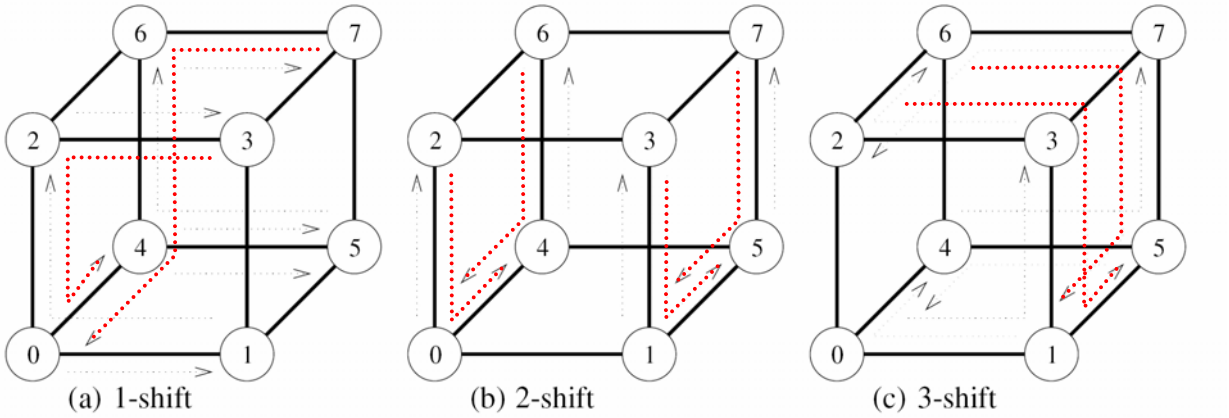
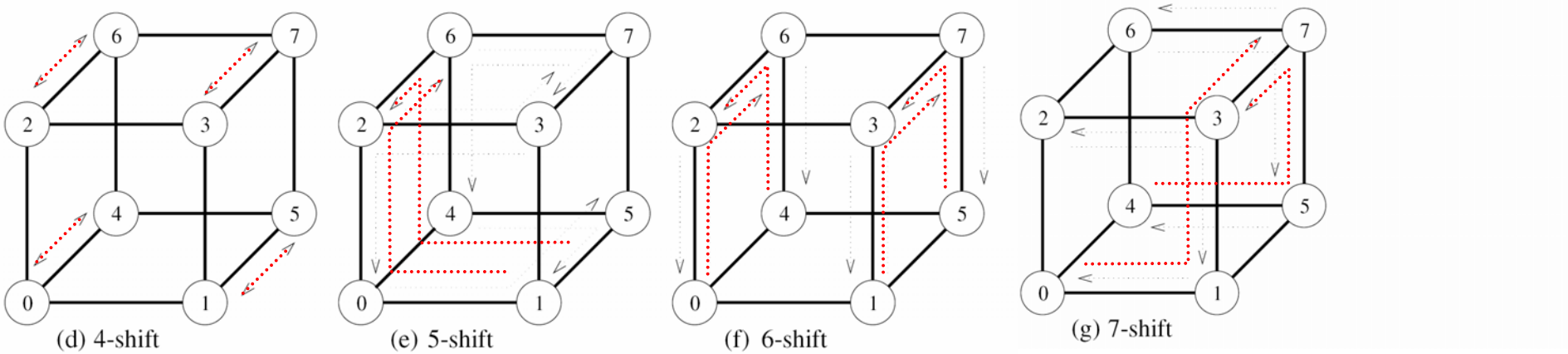
Circular q-shift - Hypercube

- Using E-cube routing $T = t_s + t_w m$
- q-shift in a hypercube with p nodes: longest path has log p - γ(q) links, where γ(q) is the highest integer j, such that q is divisable by 2^j



Pairwise communication using E-cube routing: no congestion!



Minimal cost-optimal execution time

- Isoefficiency function $\in \Theta(f(p))$:
  - A problem of size W is solved cost-optimally $\Leftrightarrow W \in \Omega(f(p))$
    p $\longrightarrow$ W grows at least like f(p)
  - or: Cost-opimality for a problem of size W $\Leftrightarrow p \in O(f^{-1}(W))$
    p grows maximally like $f^{-1}(W)$ $\longleftarrow$ W
- Parallel execution time of a cost-optimal parallel system is $\Theta(W/p)$. Furthermore it holds $1/p \in \Omega\ (1/f^{-1}(W))$.

⇨ Lower bound for parallel runtime for solving a problem of size W cost-optimally

$$T_P^{costopt} \in \Omega\left(\frac{W}{f^{-1}(W)}\right)$$

Scaled speedup

- What is the speedup S(n,p) behavior for growing values for p?

  How to choose n?

  - Memory-constrained scaled speedup:

    Memory grows propotionally with p $\Rightarrow$ determine n
    Memory requirement: $m = f_m(n)$
    Memory per node: m_0
    $f_m(n) = p\ m_0 \Rightarrow n = f_m^{-1}(p\ m_0)$

  - Time-constrained speedup:

Scaled speedup

- Example 1: Matrix-vector product:

  $T_S = t_c\, n^2$, $T_P \in \Theta(n^2/p)$, t_c : execution time for a `mult-add-operation`

$$S = \frac{t_c n^2}{t_c \frac{n^2}{p} + \underbrace{t_s \log p + t_w n}_{\text{all-to-all broadcast}}} \in \Theta(p)$$

  memory requirement: $m = f_m(n) \in \Theta(n^2)$

  Memory-constrained scaled speedup:

  available memory: $m = m_0 p \in \Theta(p)$, i.e. $n^2 = c \times p$

$$S' = \frac{t_c c \times p}{t_c \frac{c \times p}{p} + t_s \log p + t_w \sqrt{c \times p}} = \frac{c_1 p}{c_2 + c_3 \log p + c_4 \sqrt{p}} \in \Theta(\sqrt{p})$$

  Time-constrained scaled speedup:

  $T_P \in \Theta(n^2/p)$, T_P constant, i.e. $n^2 = c \times p$, scaled speedup see above

- Example 2: Matrix-Matrix multiplication

$$T_S = t_c n^3 \qquad T_P = t_c \frac{n^3}{p} + t_s \log p + 2t_w \frac{n^2}{\sqrt{p}}$$

$$S = \frac{t_c n^3}{t_c \frac{n^3}{p} + t_s \log p + 2t_w \frac{n^2}{\sqrt{p}}}$$

Memory requirement: $\Theta(n^2)$

Memory-constrained scaled speedup: $m \in \Theta(p)$, $m \in \Theta(n^2)$, i.e. $n^2 = c \times p$

$$S' = \frac{t_c (c \times p)^{1.5}}{t_c \frac{(c \times p)^{1.5}}{p} + t_s \log p + 2t_w \frac{c \times p}{\sqrt{p}}} \in \Theta(p)$$

Time-constrained scaled speedup: $T_P \in \Theta(n^3/p)$ const., i.e. $n^3 = c \times p$

$$S'' = \frac{t_c (c \times p)}{t_c \frac{c \times p}{p} + t_s \log p + 2t_w \frac{(c \times p)^{2/3}}{\sqrt{p}}} \in \Theta(p^{5/6})$$