New link arrangements for Dragonfly networks

Madison Belka	Knox College
Myra Doubet	Knox College
Sofia Meyers	Knox College
Rosemary Momoh	Knox College
David Rincon-Cruz	Knox College (now Columbia Univ.)
David P. Bunde	Knox College

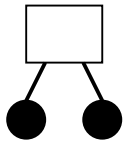
This work was partially supported by the National Science Foundation under grant CNS-1423413. The views expressed are the presenter's.

Dragonfly

- Hierarchical architecture to exploit high-radix switches and optical links

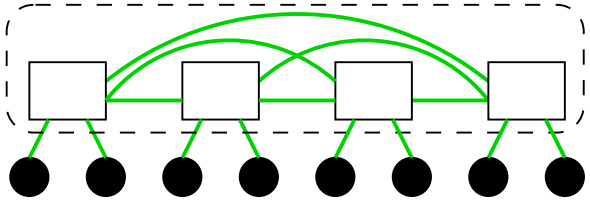
Dragonfly

- Hierarchical architecture to exploit high-radix switches and optical links
  - Nodes attached to switches



Dragonfly

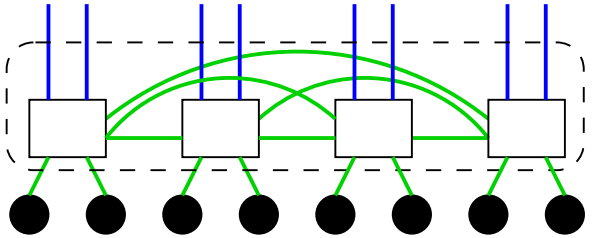
- Hierarchical architecture to exploit high-radix switches and optical links
  - Nodes attached to switches
  - Switches form groups



  - Group members connected w/ local edge (electrical)

Dragonfly

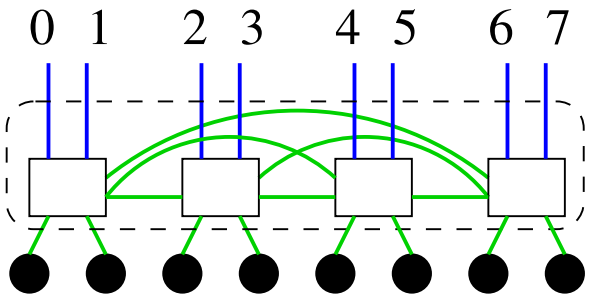
- Hierarchical architecture to exploit high-radix switches and optical links
  - Nodes attached to switches
  - Switches form groups



  - Group members connected w/ local edge (electrical)
  - Each pair of groups connected w/ global edge (optical)

Dragonfly

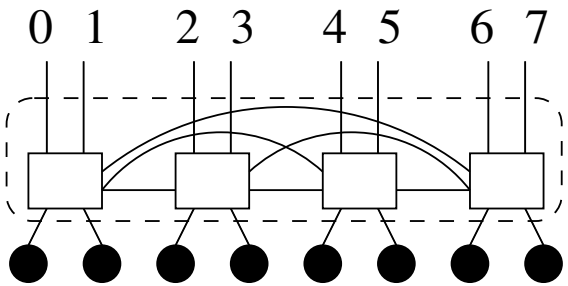
- Hierarchical architecture to exploit high-radix switches and optical links
  - Nodes attached to switches
  - Switches form groups



  - Group members connected w/ local edge (electrical)
  - Each pair of groups connected w/ global edge (optical)

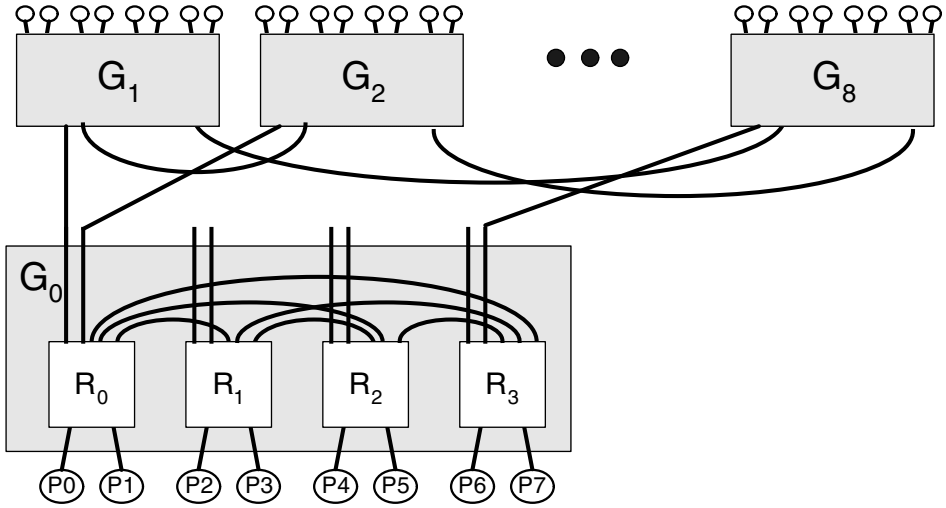
Dragonfly parameters

- p = number of nodes connected to a switch
- a = number of switches in a group
- h = number of optical links on a switch



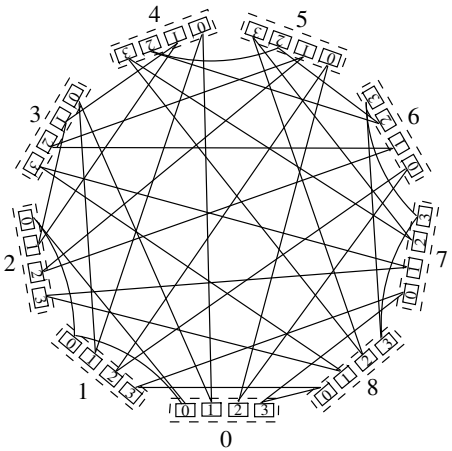
- Number of groups g = ah+1

Which port connects to which group?

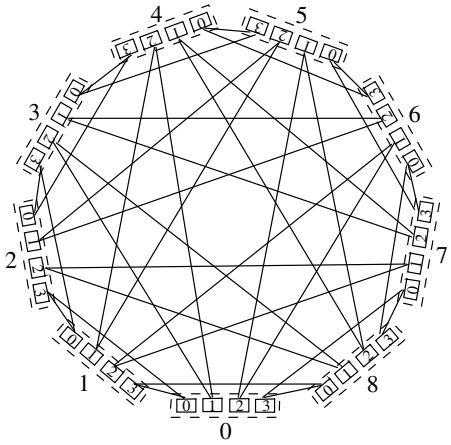


From original Dragonfly paper: Kim et al., ISCA 2008

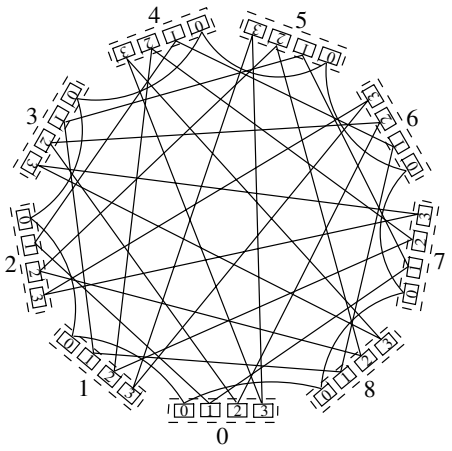
Previously known: Three distinct global link arrangements



Absolute arrangement



Relative arrangement



Circulant-based arrangement

Arrangements defined in Camarero et al. ACM Trans. Architec. Code Optim., 2014.

Note:

IBM implementation (PERCS) uses absolute

Researchers who draw entire system in their papers use relative

Bisection bandwidth

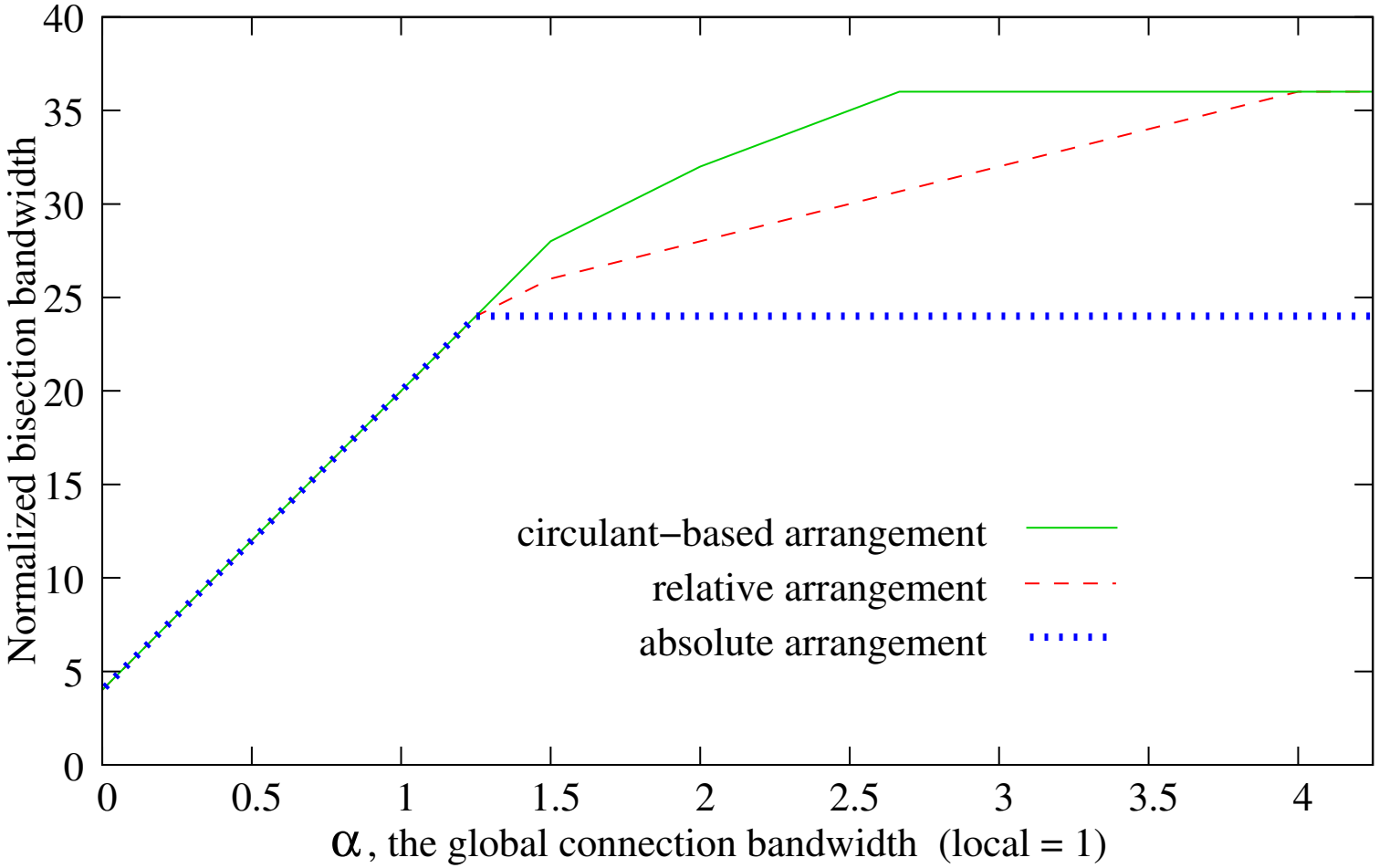
- Minimum bandwidth between two equal-sized parts of the system
  - Bandwidth for a particular bisection is the (weighted) number of edges crossing from one part to the other
  - Minimize this over all bisections
- Tries to measure worst-case communication bottleneck in a large computation

Bisection bandwidth

- Minimum bandwidth between two equal-sized parts of the system
  - Bandwidth for a particular bisection is the (weighted) number of edges crossing from one part to the other
  - Minimize this over all bisections
- Tries to measure worst-case communication bottleneck in a large computation
- We treat local and global edges differently
  - local edge weights to 1
  - global edge weights to α

Arrangements give different bisection BW

[Hastings et al., Cluster 2015]



Bisection bandwidth as function of α for (p,4,2)-Dragonfly

Flavor of results for large networks

[Hastings et al, Cluster 2015]

- Bisection bandwidth for relative arrangement:

  $(a/2)^2 g$ if $a \bmod 4 = 0$ and α is large

  $\Theta(\alpha)$ if $a \bmod 4 \neq 0$

Flavor of results for large networks

[Hastings et al, Cluster 2015]

- Bisection bandwidth for relative arrangement:

  $(a/2)^2 g$ if $a \bmod 4 = 0$ and α is large

  $\Theta(\alpha)$ if $a \bmod 4 \neq 0$

- *Globally connected component* (GCC): A connected component of the network with only global links (ignoring local links)

Our question

- Can we make a global link arrangement that forms a single GCC? How does it perform?

Our question

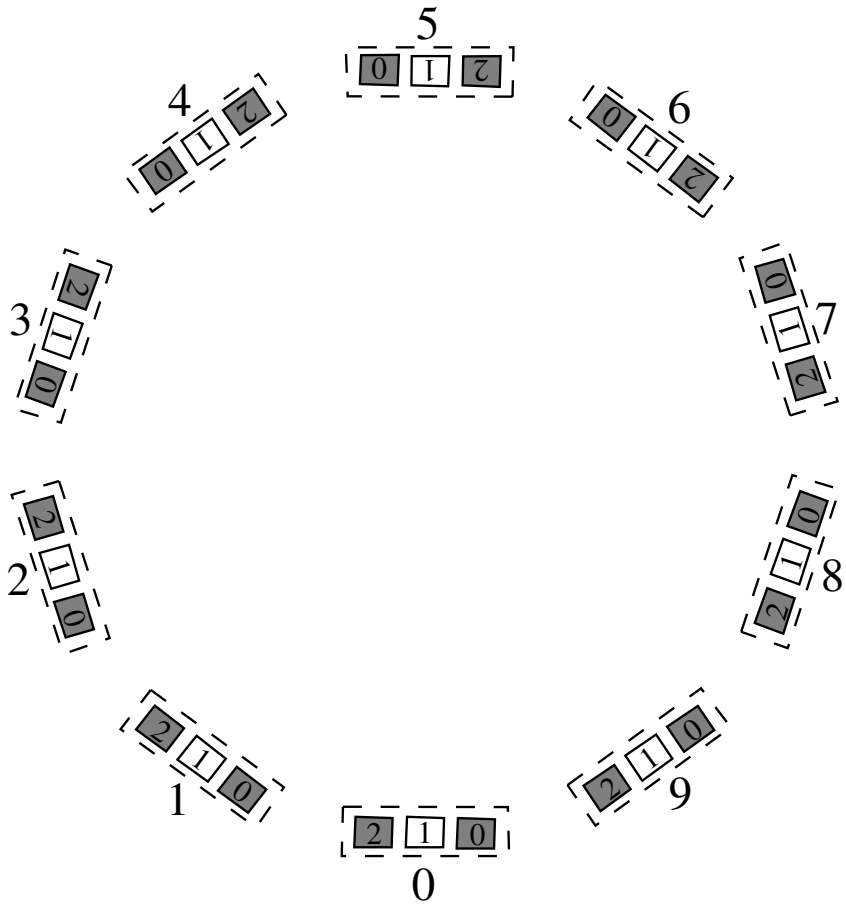
- Can we make a global link arrangement that forms a single GCC? How does it perform?

  Yes – we made 2 of them (Nautilus and Helix)

  Their bisection bandwidth is
  - generally better at high α
  - and at least as good for low α

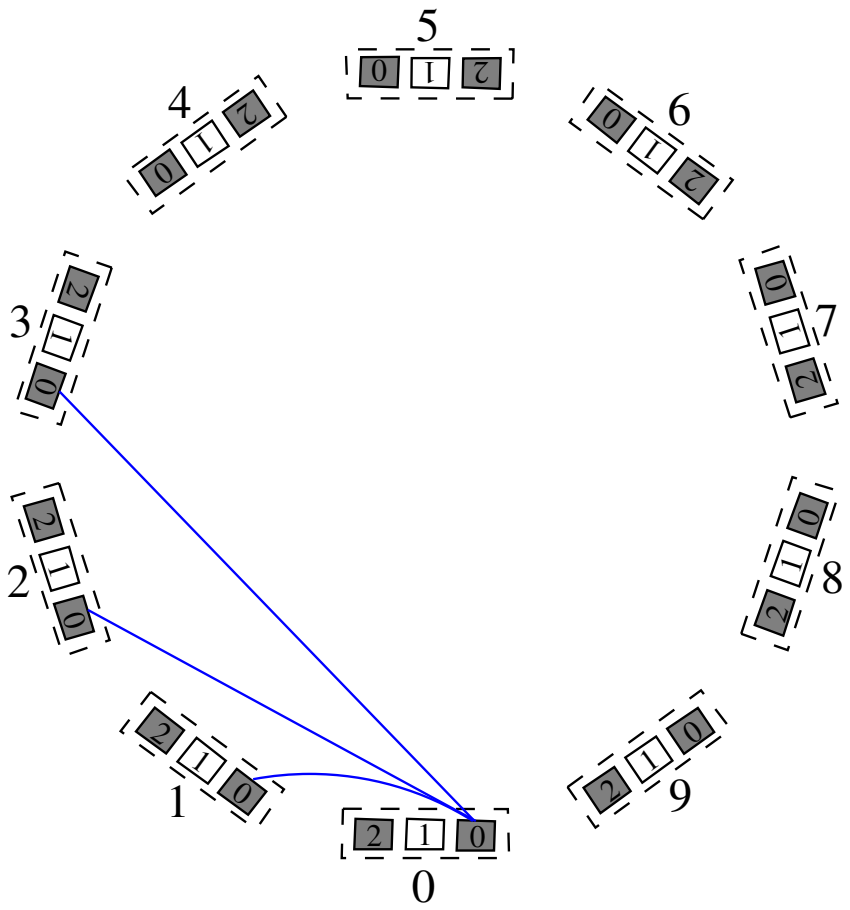
Nautilus global link arrangement

- Mark even switches (shaded). These go CW.



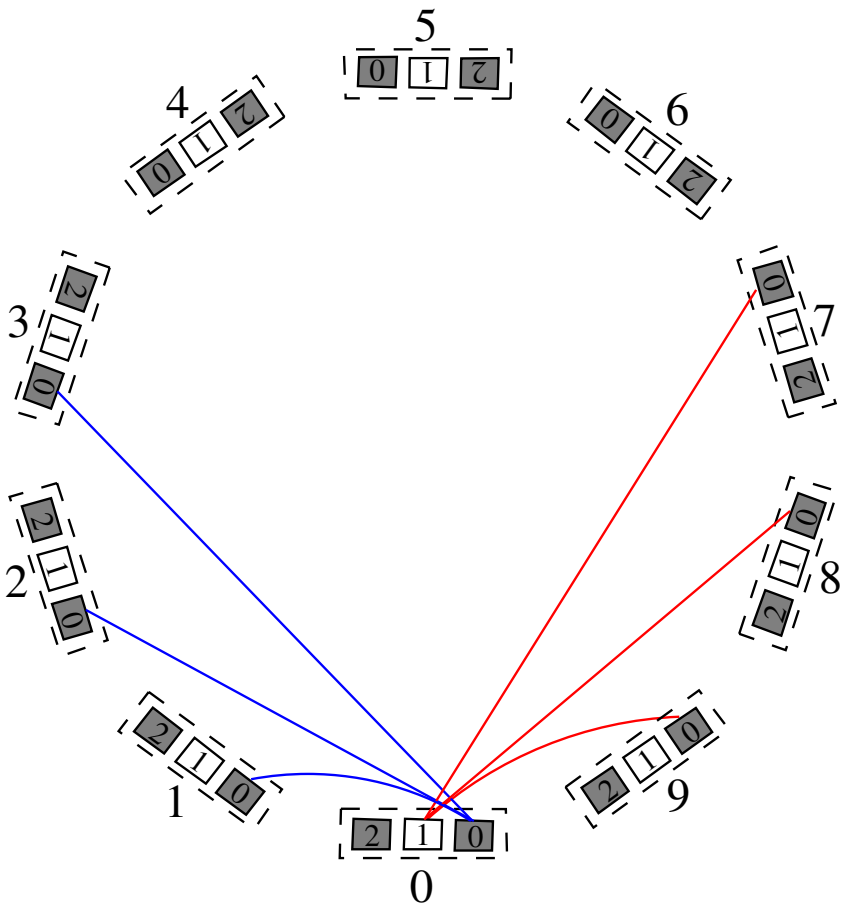
Nautilus global link arrangement

- Mark even switches (shaded). These go CW.
- Visit each switch in turn
  - Add remaining edges to "next" groups in its direction
  - Edges from group i connect to switch i % a in destination group



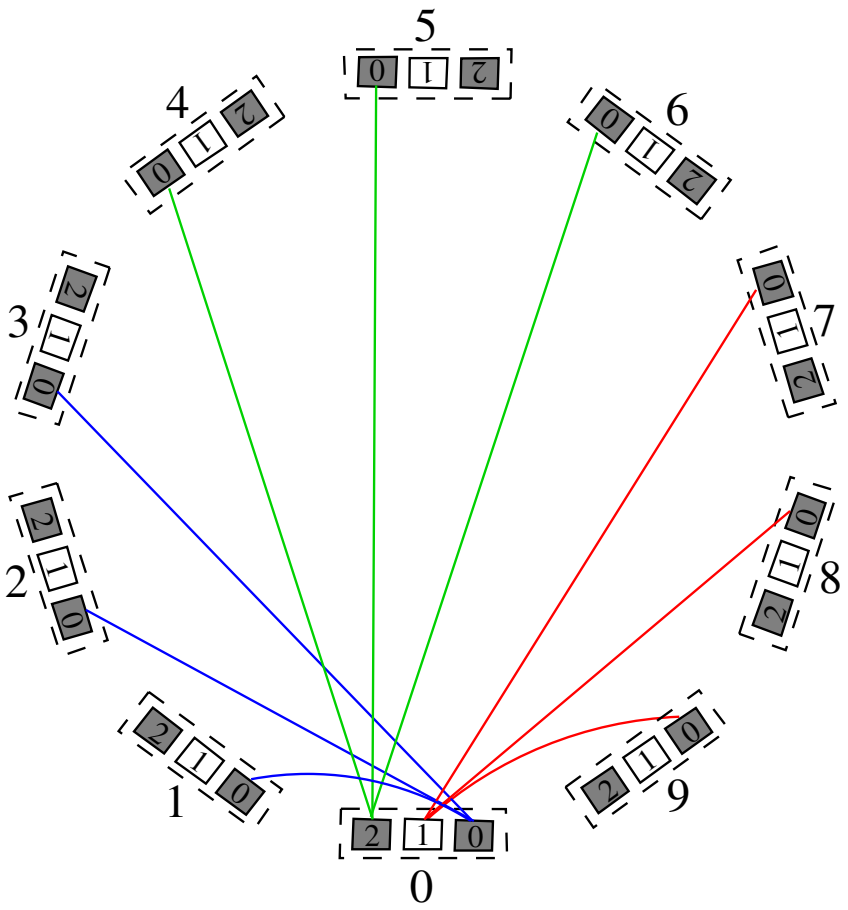
Nautilus global link arrangement

- Mark even switches (shaded). These go CW.
- Visit each switch in turn
  - Add remaining edges to "next" groups in its direction
  - Edges from group i connect to switch i % a in destination group



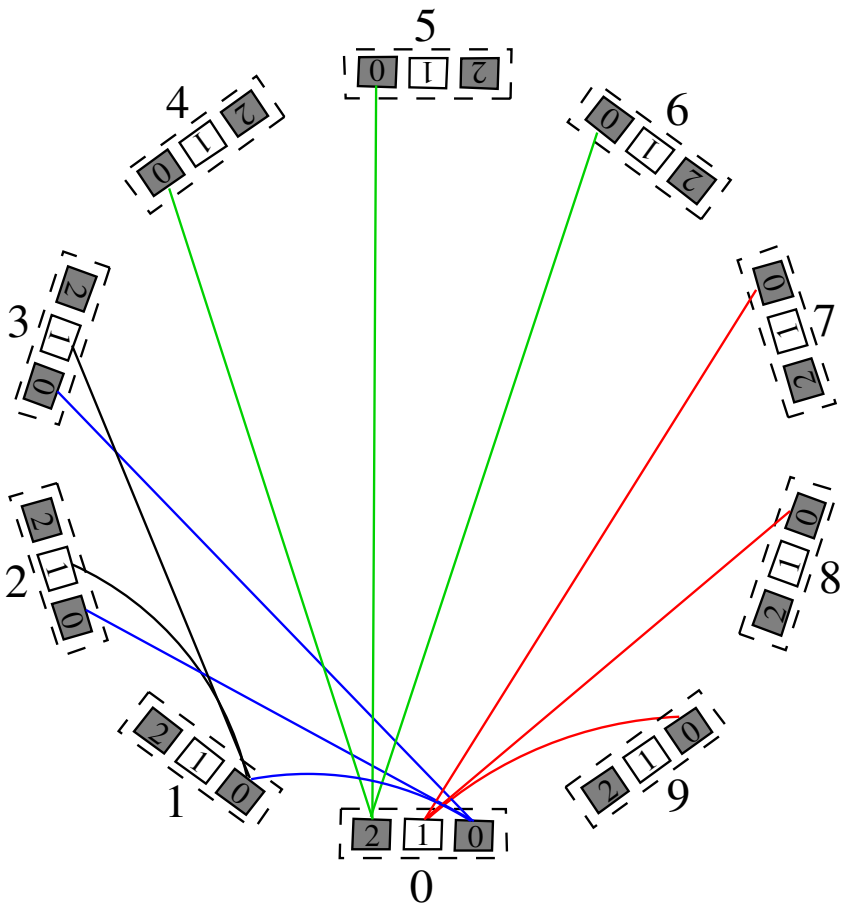
Nautilus global link arrangement

- Mark even switches (shaded). These go CW.
- Visit each switch in turn
  - Add remaining edges to "next" groups in its direction
  - Edges from group i connect to switch i % a in destination group



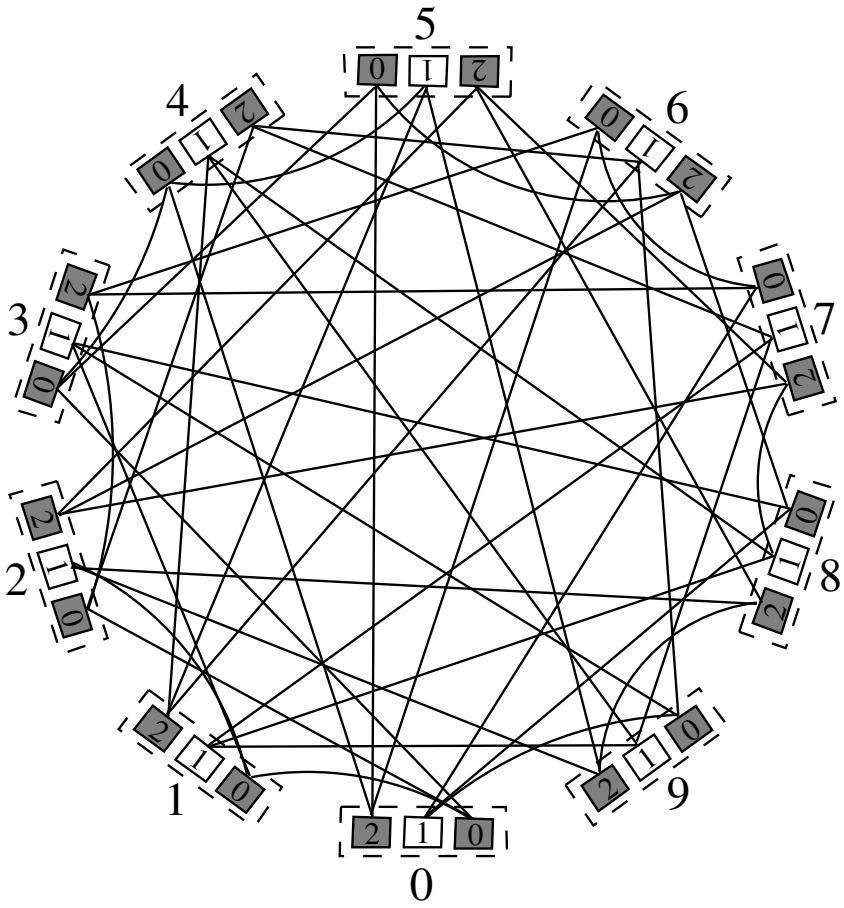
Nautilus global link arrangement

- Mark even switches (shaded). These go CW.
- Visit each switch in turn
  - Add remaining edges to "next" groups in its direction
  - Edges from group i connect to switch i % a in destination group



Nautilus global link arrangement

- Mark even switches (shaded). These go CW.
- Visit each switch in turn
  - Add remaining edges to "next" groups in its direction
  - Edges from group i connect to switch i % a in destination group



Results on Nautilus arrangement

- Each pair of groups is connected by exactly 1 link and every node has h links

Results on Nautilus arrangement

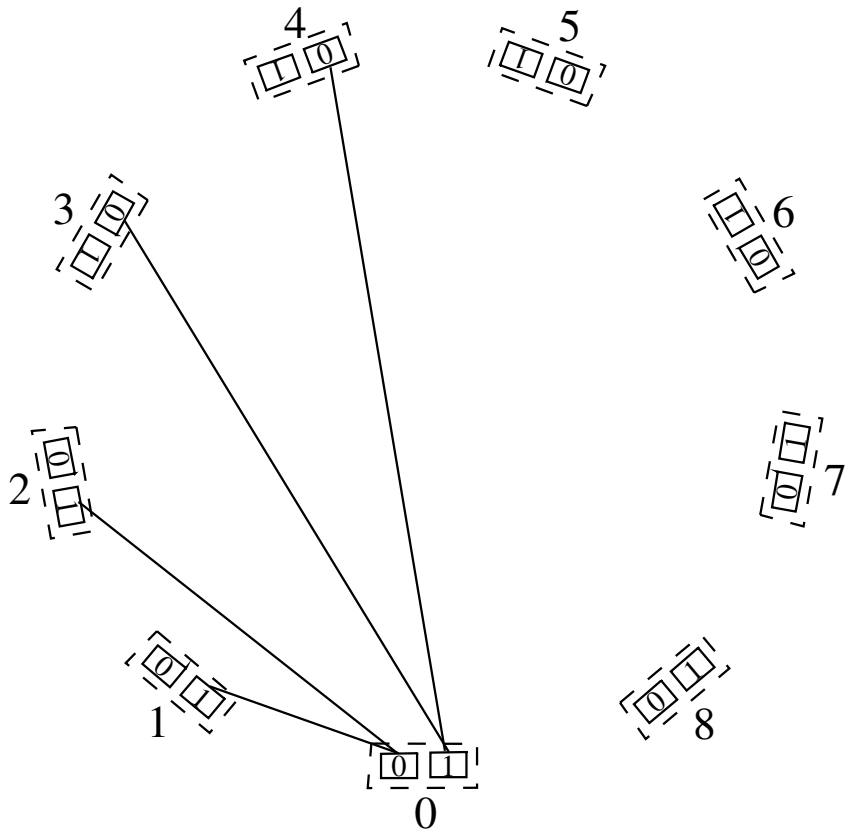
- Each pair of groups is connected by exactly 1 link and every node has h links
- Closed form formula for which pairs of nodes are connected

Results on Nautilus arrangement

- Each pair of groups is connected by exactly 1 link and every node has h links
- Closed form formula for which pairs of nodes are connected
- 1 GCC is formed when $h > 2$ and
    i. $a < h$,
    ii. $a = h$, or
    iii. $a = 2h$

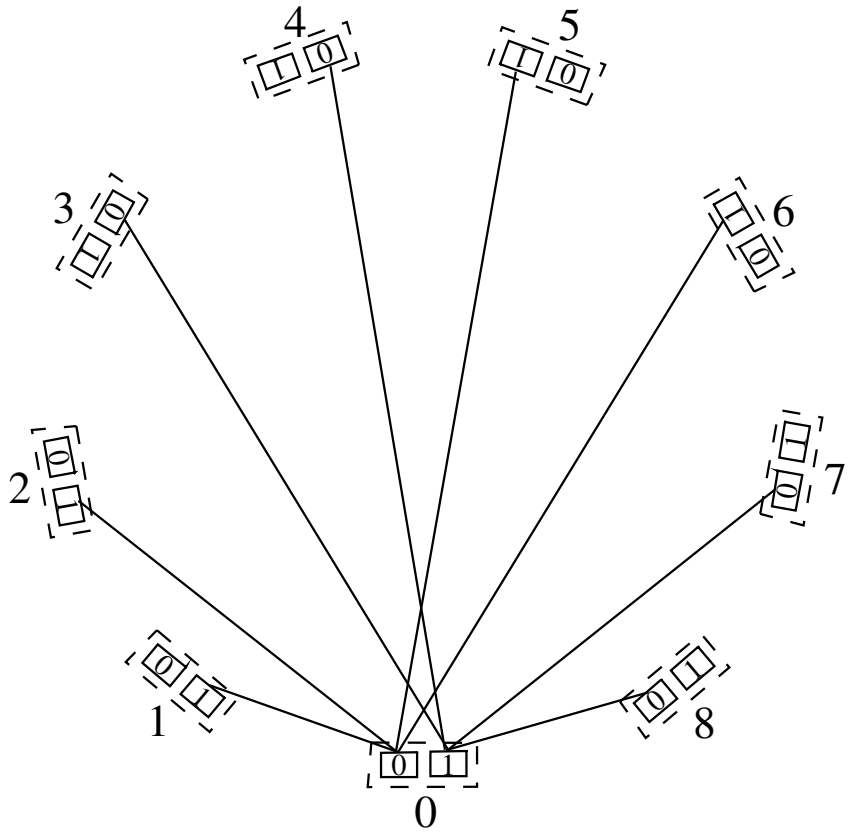
Helix global link arrangement

- If h is even, divide links into h/2 outgoing and h/2 incoming
- Outgoing links go to next h/2 groups, one switch higher



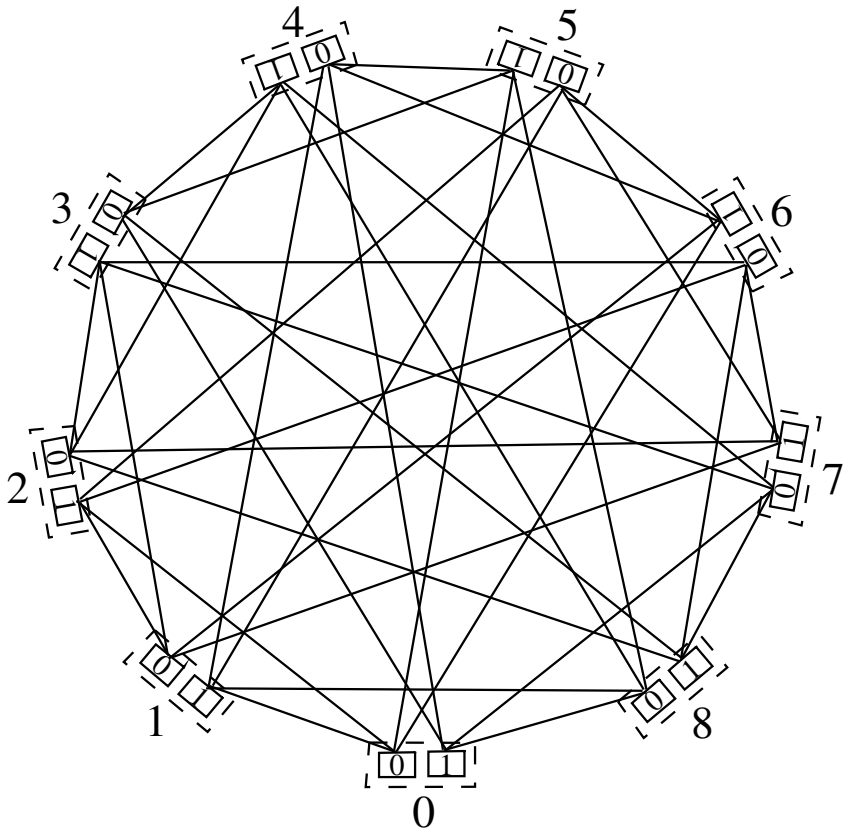
Helix global link arrangement

- If h is even, divide links into h/2 outgoing and h/2 incoming
- Outgoing links go to next h/2 groups, one switch higher



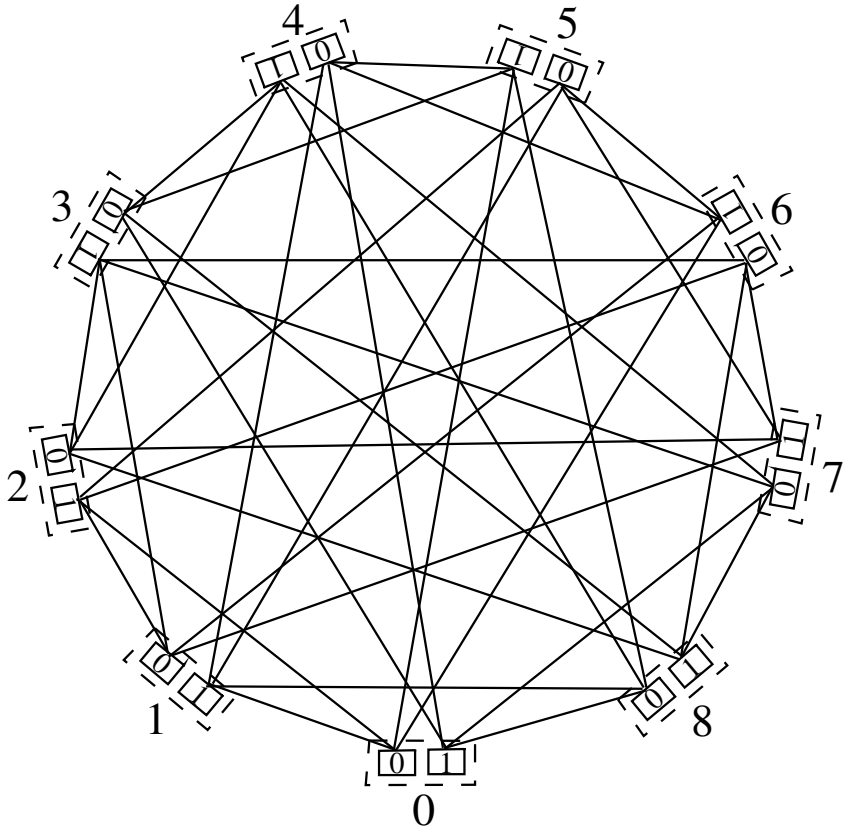
Helix global link arrangement

- If h is even, divide links into h/2 outgoing and h/2 incoming
- Outgoing links go to next h/2 groups, one switch higher



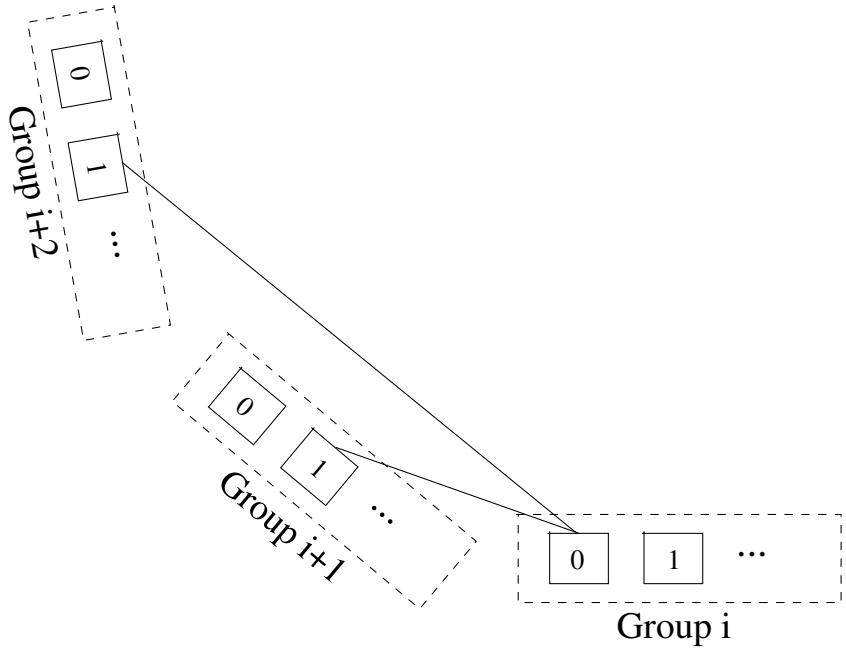
Helix global link arrangement

- If h is even, divide links into h/2 outgoing and h/2 incoming
- Outgoing links go to next h/2 groups, one switch higher
- If h is odd, the “middle links” of each switch go to uncovered groups



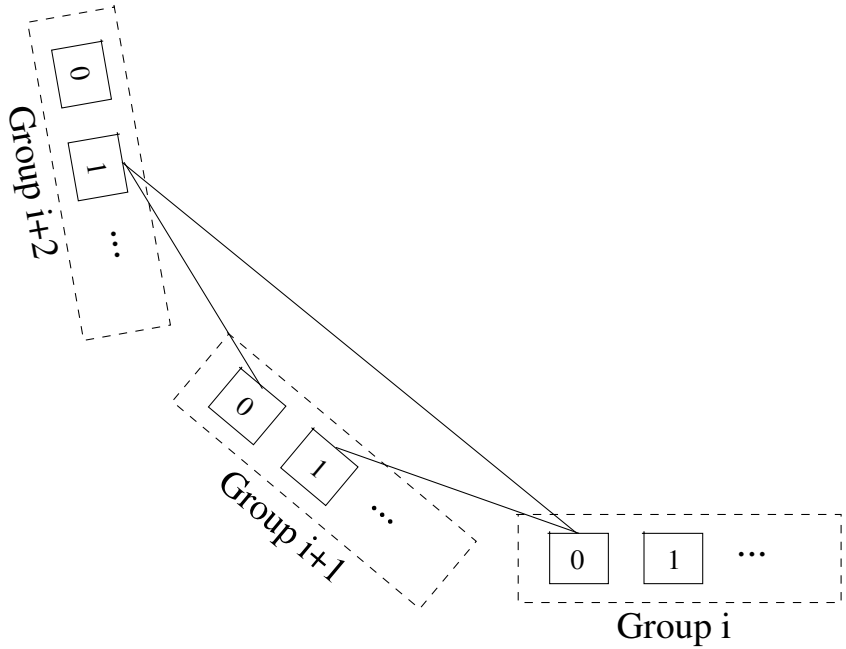
Helix arrangement forms 1 GCC
(when h ≥ 4)

- Group i, switch 0 connects to switch 1 of group i+2



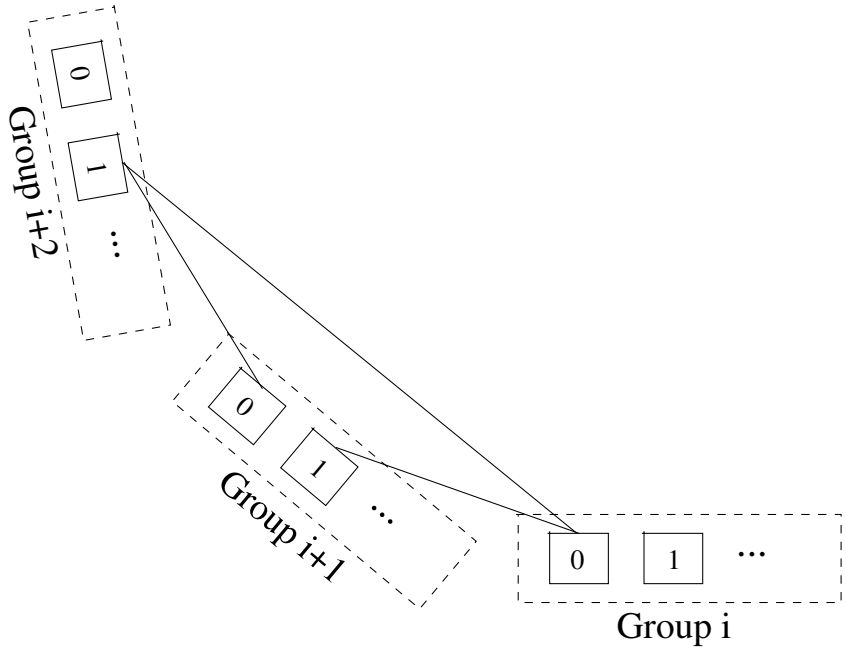
Helix arrangement forms 1 GCC
(when h ≥ 4)

- Group i, switch 0 connects to switch 1 of group i+2
- Group i, switch 0 connects to same switch



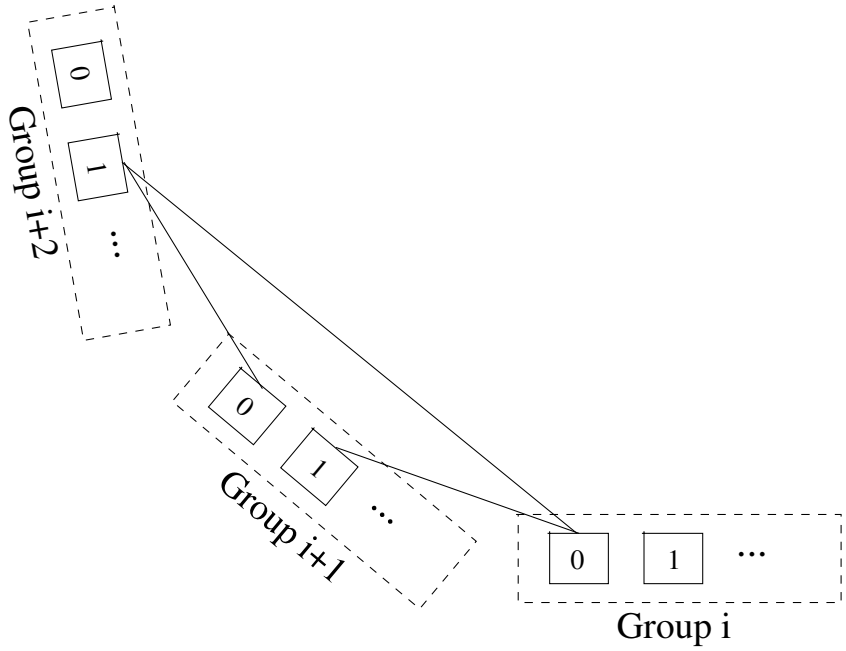
Helix arrangement forms 1 GCC
(when h ≥ 4)

- Group i, switch 0 connects to switch 1 of group i+2
- Group i, switch 0 connects to same switch
- Therefore all 0 switches are connected



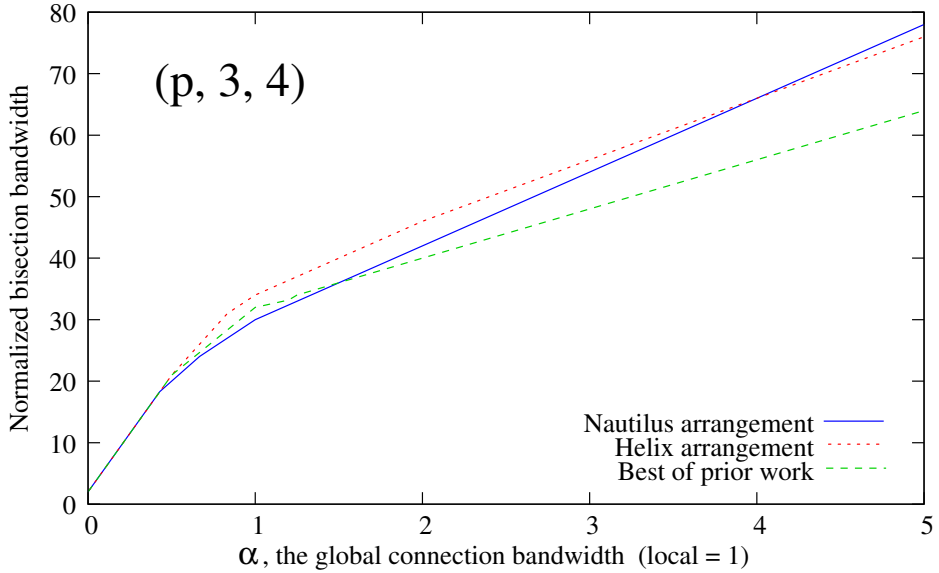
Helix arrangement forms 1 GCC
(when h ≥ 4)

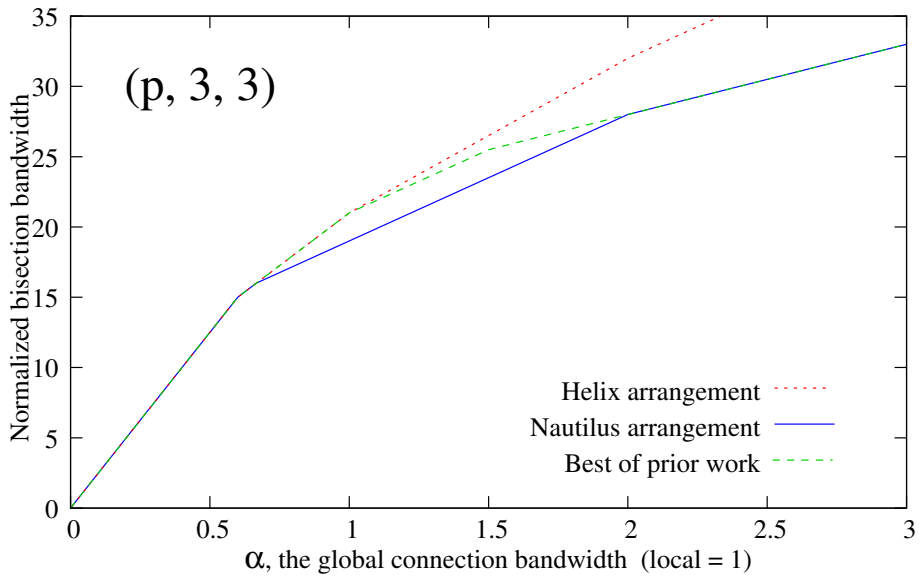
- Group i, switch 0 connects to switch 1 of group i+2
- Group i, switch 0 connects to same switch
- Therefore all 0 switches are connected
- Therefore all switches are connected

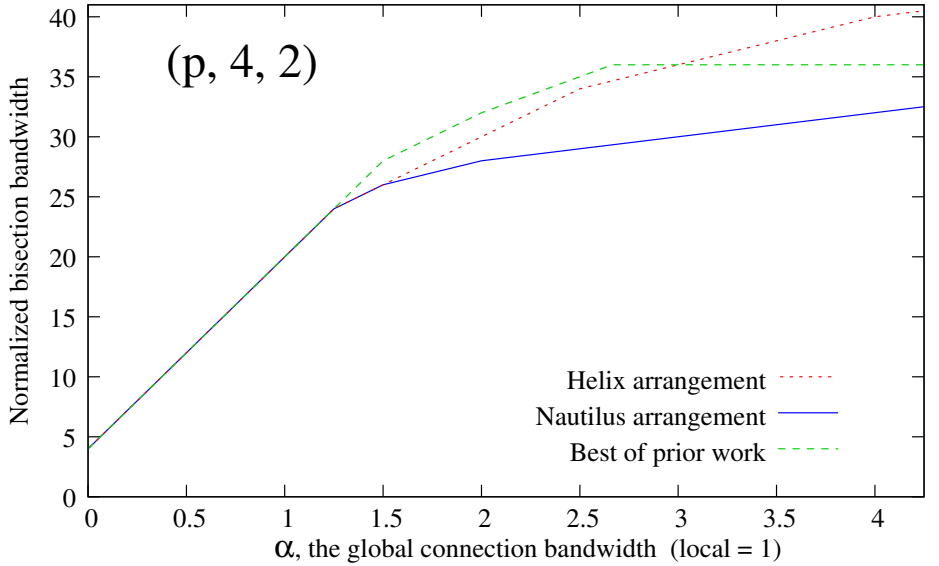


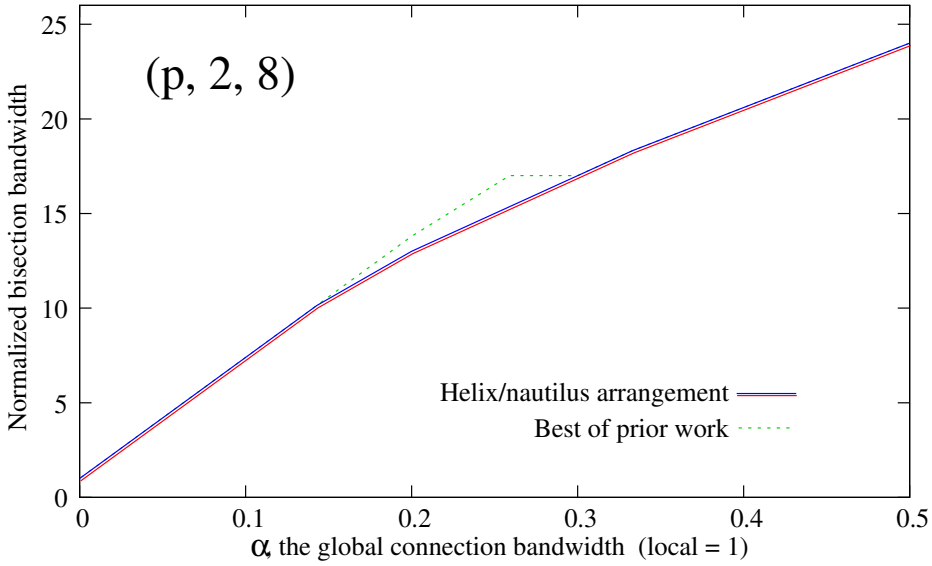
Bisection bandwidth on small networks

(p, a, h) = (nodes/switch, switches/group, links/switch)









Conclusions

- New arrangements
  - Better at large α
  - At least as good for small α
  - Sometimes inferior at intermediate α
- The symmetry of Helix seems to make it preferable to Nautilus

Future work

- What is relationship between bisection bandwidth results and empirical network performance?

- Remaining cases for large α and exact values for general network sizes

Thanks!

dbunde@knox.edu