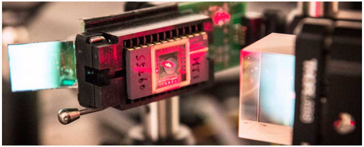
Routing Schemes for Hybrid Communication Networks

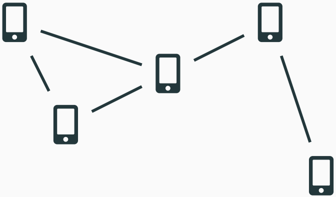
SIROCCO 2023

Sam Coy, Artur Czumaj, Christian Scheideler, Philipp Schneider, **Julian Werthmann**

June 7, 2023

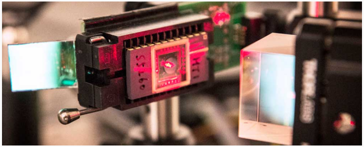
Model & Motivation - Hybrid Communication

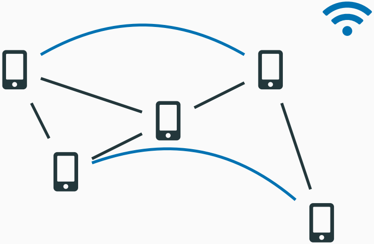




- Communicate via different channels at the same time
    - Data Centers (Cables + Lasers)
    - Wireless Networks (Ad hoc + Cellular)

Model & Motivation - Hybrid Communication





- Communicate via different channels at the same time
    - Data Centers (Cables + Lasers)
    - Wireless Networks (Ad hoc + Cellular)

Model & Motivation - Hybrid Communication

- **HYBRID** model[1]
  - Synchronous rounds
  - Local edges: **CONGEST**
    - Send one message of size $\mathcal{O}(\log n)$ per neighbor per round
  - Global edges: $\mathbf{NCC_0}$[2]
    - Send & Receive $\mathcal{O}(\log n)$ messages of size $\mathcal{O}(\log n)$ per round
    - Can send messages only if target known

[1] Augustine et al., "Shortest Paths in a Hybrid Network Model".
[2] Augustine et al., "Distributed Graph Realizations †".

Model & Motivation - Hybrid Communication

- **HYBRID** model[1]
  - Synchronous rounds
  - Local edges: **CONGEST**
    - Send one message of size $\mathcal{O}(\log n)$ per neighbor per round
  - Global edges: $\mathbf{NCC_0}$[2]
    - Send & Receive $\mathcal{O}(\log n)$ messages of size $\mathcal{O}(\log n)$ per round
    - Can send messages only if target known
- n nodes with unique identifiers and positions in $\mathbb{Z}^2$
- Grid graph $G = (V, E)$, $\{v, w\} \in E \Leftrightarrow \|v - w\|_2 = 1$

[1] Augustine et al., "Shortest Paths in a Hybrid Network Model".
[2] Augustine et al., "Distributed Graph Realizations †".

Model & Motivation - Problem Definition

- **Preprocessing Phase:** Compute routing tables and node labels for each node
- **Routing Phase:** Given target node's label forward packet using routing table

Model & Motivation - Problem Definition

- **Preprocessing Phase:** Compute routing tables and node labels for each node
- **Routing Phase:** Given target node's label forward packet using routing table

Goals

Routing tables and node labels for **local graph** with:

Model & Motivation - Problem Definition

- **Preprocessing Phase:** Compute routing tables and node labels for each node
- **Routing Phase:** Given target node's label forward packet using routing table

Goals

Routing tables and node labels for **local graph** with:

- Fast preprocessing

Model & Motivation - Problem Definition

- **Preprocessing Phase:** Compute routing tables and node labels for each node
- **Routing Phase:** Given target node's label forward packet using routing table

Goals

Routing tables and node labels for **local graph** with:

- Fast preprocessing
- Small node labels

Model & Motivation - Problem Definition

- **Preprocessing Phase:** Compute routing tables and node labels for each node
- **Routing Phase:** Given target node's label forward packet using routing table

Goals

Routing tables and node labels for **local graph** with:

- Fast preprocessing
- Small routing tables
- Small node labels

Model & Motivation - Problem Definition

- **Preprocessing Phase:** Compute routing tables and node labels for each node
- **Routing Phase:** Given target node's label forward packet using routing table

Goals

Routing tables and node labels for **local graph** with:

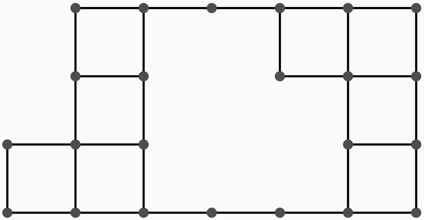
- Fast preprocessing
- Small routing tables
- Small node labels
- Small stretch

Related Work - No Radio Holes[3]

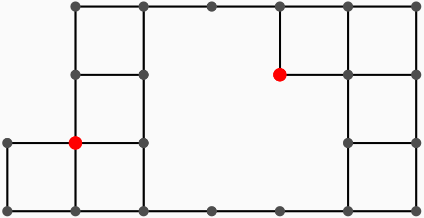
- $\mathcal{O}(\log n)$ rounds of preprocessing
- Node labels of size $\mathcal{O}(\log n)$
- $\mathcal{O}(\log n)$ bits of information stored at each node
- Exact in grid graphs, constant stretch in UDGs

[3] Coy et al., *Near-Shortest Path Routing in Hybrid Communication Networks*.

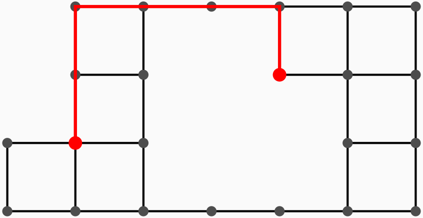
Contributions - Radio Holes



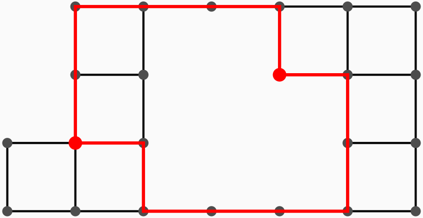
Contributions - Radio Holes



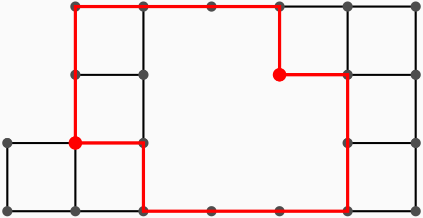
Contributions - Radio Holes



Contributions - Radio Holes

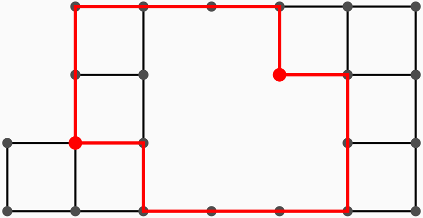


Contributions - Radio Holes



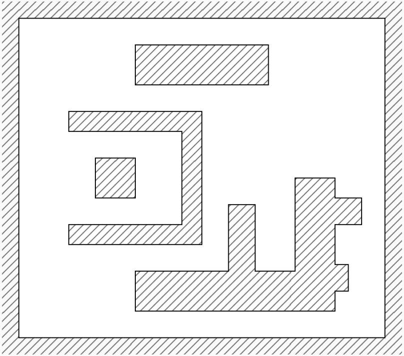
- Paths can no longer be transformed into each other

Contributions - Radio Holes



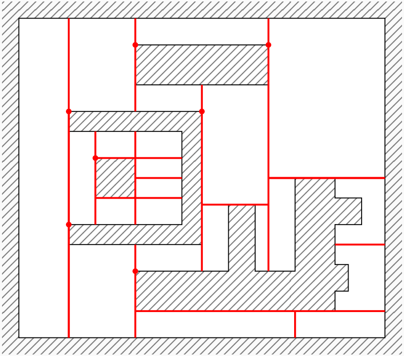
- Paths can no longer be transformed into each other
- Number of classes of paths scale fast with number of holes

Contributions - Approach



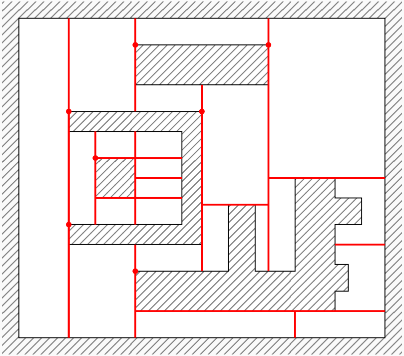
Contributions - Approach

- Divide grid graph into regions



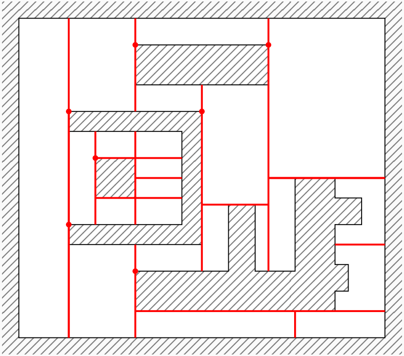
Contributions - Approach

- Divide grid graph into regions
- Use related work's scheme to route inside regions



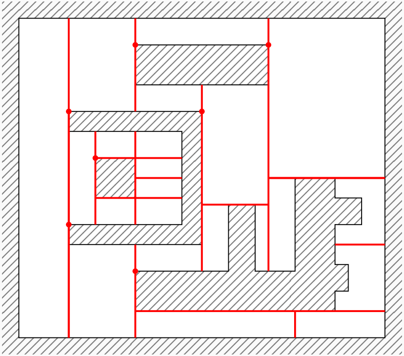
Contributions - Approach

- Divide grid graph into regions
- Use related work's scheme to route inside regions
- Use new scheme to route towards target region



Contributions - Approach

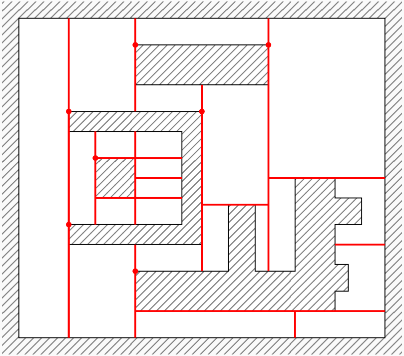
- Divide grid graph into regions
- Use related work's scheme to route inside regions
- Use new scheme to route towards target region



Regionalization Requirements

Contributions - Approach

- Divide grid graph into regions
- Use related work's scheme to route inside regions
- Use new scheme to route towards target region

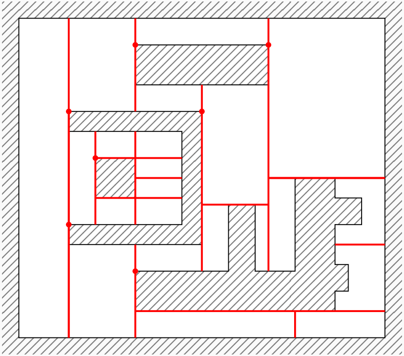


Regionalization Requirements

- Simple: The regions contain no holes

Contributions - Approach

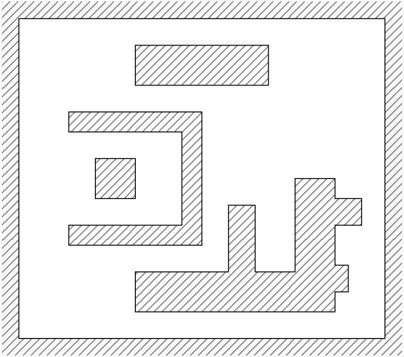
- Divide grid graph into regions
- Use related work's scheme to route inside regions
- Use new scheme to route towards target region



Regionalization Requirements

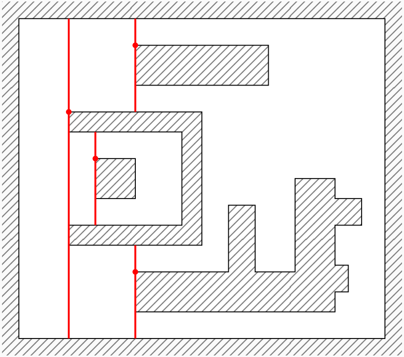
- Simple: The regions contain no holes
- Path-convex: For each pair of nodes in a region, there is a shortest path inside that region

Contributions - Regionalization Steps



Contributions - Regionalization Steps

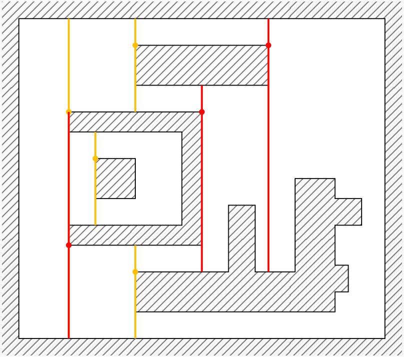
I: Simple Regions



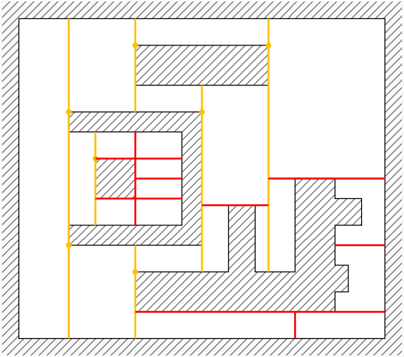
Contributions - Regionalization Steps

I: Simple Regions

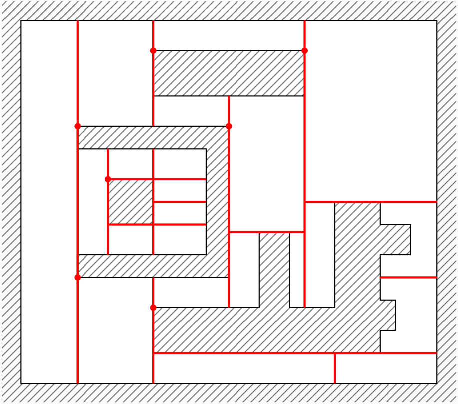
II: Tunnels



- I: Simple Regions
- II: Tunnels
- III: Path-Convex Regions



Contributions - Regionalization Result



Runtime: $\mathcal{O}(\log n)$ #Regions: $\mathcal{O}(h)$

Contributions - Region Routing

- Problem: How to decide which region to go to next?
- Solution: Landmark Graph

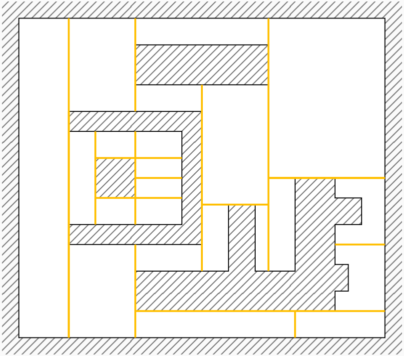
Contributions - Region Routing

- Problem: How to decide which region to go to next?
- Solution: Landmark Graph
- Shortest path in Landmark Graph corresponds to shortest path through regions

Contributions - Region Routing

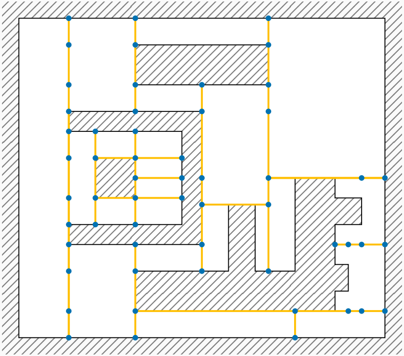
- Problem: How to decide which region to go to next?
- Solution: Landmark Graph
- Shortest path in Landmark Graph corresponds to shortest path through regions
- Making it part of routing table allows local decisions

Contributions - Landmark Graph



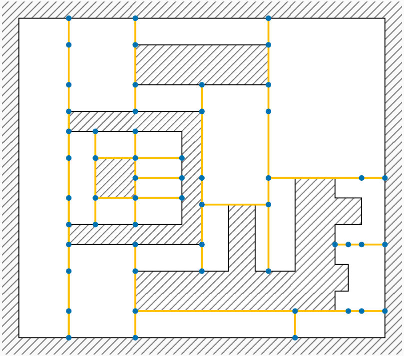
Contributions - Landmark Graph

- Mark key nodes of the graph as landmarks



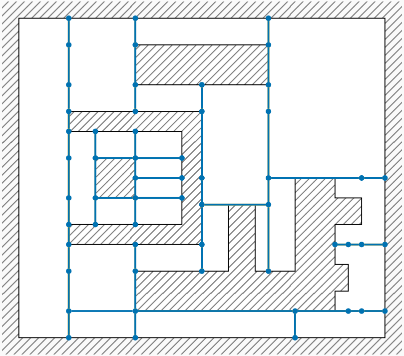
Contributions - Landmark Graph

- Mark key nodes of the graph as landmarks
- Connect two landmarks, if



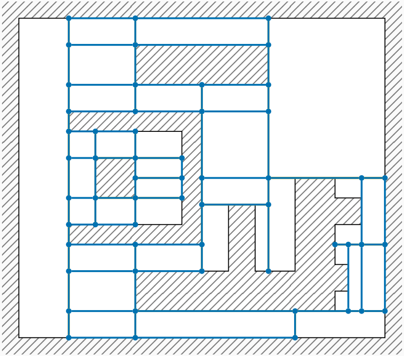
Contributions - Landmark Graph

- Mark key nodes of the graph as landmarks
- Connect two landmarks, if
    - Adjacent on same gate



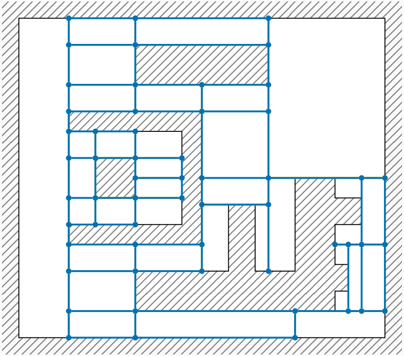
Contributions - Landmark Graph

- Mark key nodes of the graph as landmarks
- Connect two landmarks, if
  - Adjacent on same gate
  - On Adjacent gates and closest

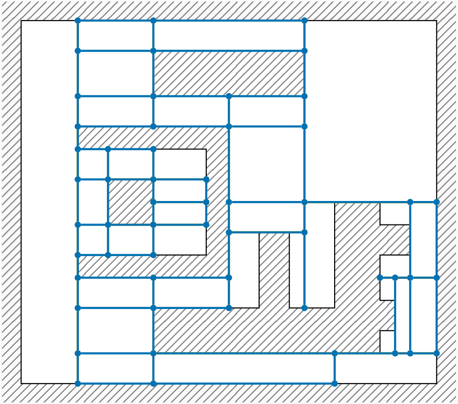


Contributions - Landmark Graph

- Mark key nodes of the graph as landmarks
- Connect two landmarks, if
  - Adjacent on same gate
  - On Adjacent gates and closest
- Add distances as weights

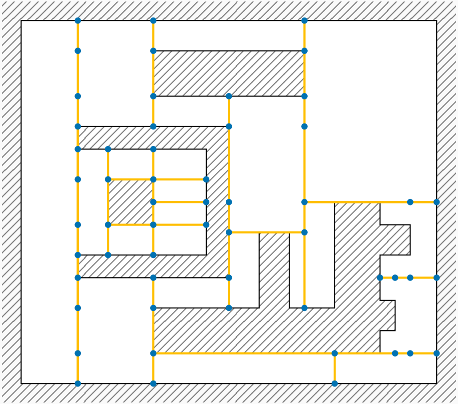


Contributions - Landmark Graph Result



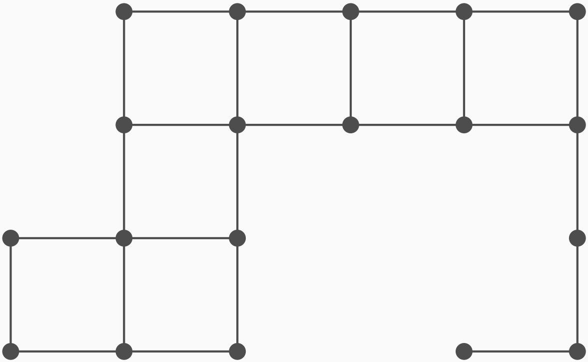
Runtime: $\mathcal{O}(\log n)$ #Landmarks: $\mathcal{O}(h^2)$ #Edges: $\mathcal{O}(h^2)$

Contributions - Landmark Graph Result

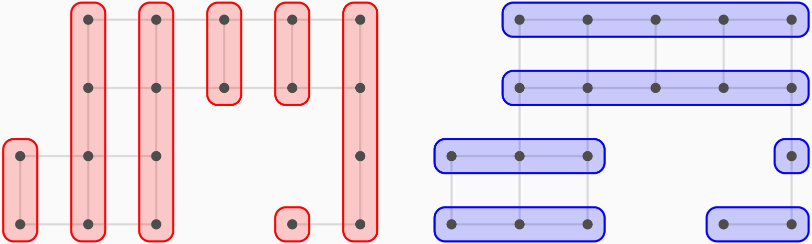


Runtime: $\mathcal{O}(\log n)$ #Landmarks: $\mathcal{O}(h^2)$ #Edges: $\mathcal{O}(h^2)$

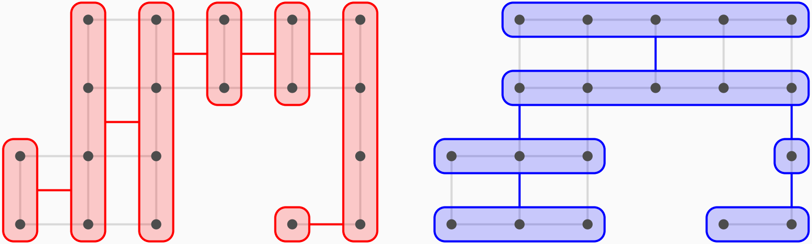
Contributions - SSSP without holes



Contributions - SSSP without holes

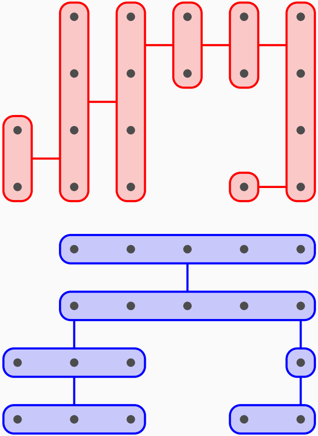


Contributions - SSSP without holes



Contributions - SSSP without holes

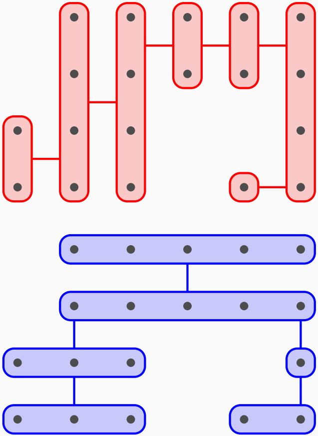
- Solve SSSP in both trees[4]



[4] Feldmann, Hinnenthal, and Scheideler, "Fast Hybrid Network Algorithms for Shortest Paths in Sparse Graphs".

Contributions - SSSP without holes

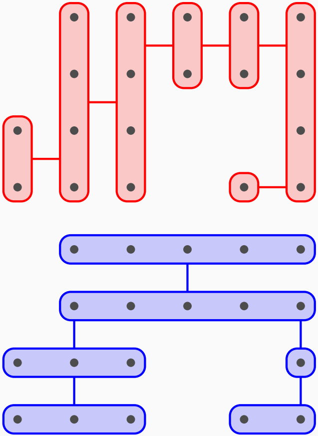
- Solve SSSP in both trees[4]
- Vertical: Amount of horizontal steps



[4] Feldmann, Hinnenthal, and Scheideler, "Fast Hybrid Network Algorithms for Shortest Paths in Sparse Graphs".

Contributions - SSSP without holes

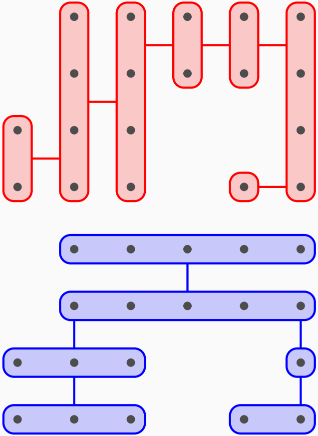
- Solve SSSP in both trees[4]
- Vertical: Amount of horizontal steps
- Horizontal: Amount of vertical steps



[4] Feldmann, Hinnenthal, and Scheideler, "Fast Hybrid Network Algorithms for Shortest Paths in Sparse Graphs".

Contributions - SSSP without holes

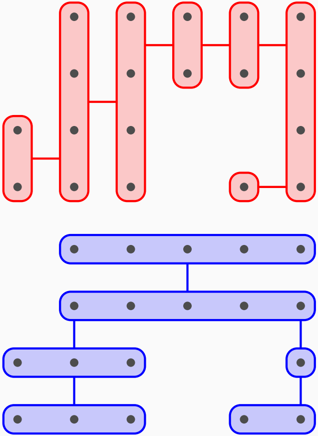
- Solve SSSP in both trees[4]
- Vertical: Amount of horizontal steps
- Horizontal: Amount of vertical steps
- Sum: Total distance



[4] Feldmann, Hinnenthal, and Scheideler, "Fast Hybrid Network Algorithms for Shortest Paths in Sparse Graphs".

Contributions - SSSP without holes

- Solve SSSP in both trees[4]
- Vertical: Amount of horizontal steps
- Horizontal: Amount of vertical steps
- Sum: Total distance
- Runtime: $\mathcal{O}(\log n)$



[4] Feldmann, Hinnenthal, and Scheideler, "Fast Hybrid Network Algorithms for Shortest Paths in Sparse Graphs".

Contributions - Preprocessing Wrapup

Preprocessing Step	Runtime
Regionalization	$\mathcal{O}(\log n)$
Computing landmark graph	$\mathcal{O}(\log n)$
Distributing landmark graph[5]	$\mathcal{O}(h^2 + \log n)$
SSSP from each landmark	$\mathcal{O}(\log n)$
'SSSP' from each gate	$\mathcal{O}(\log n)$
Distributing region indentifiers	$\mathcal{O}(\log n)$
Region routing tables[6]	$\mathcal{O}(\log n)$
Total	$\mathcal{O}(h^2 + \log n)$

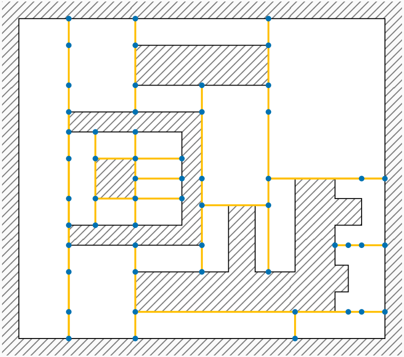
[5] Augustine et al., "Distributed Computation in Node-Capacitated Networks".
[6] Coy et al., *Near-Shortest Path Routing in Hybrid Communication Networks*.

Contributions - Node Labels and Routing Tables

Node Label Information	Bits
Node identifier	$\mathcal{O}(\log n)$
Region identifer	$\mathcal{O}(\log n)$
Region distance information	$\mathcal{O}(\log n)$
Total	$\mathcal{O}(\log n)$

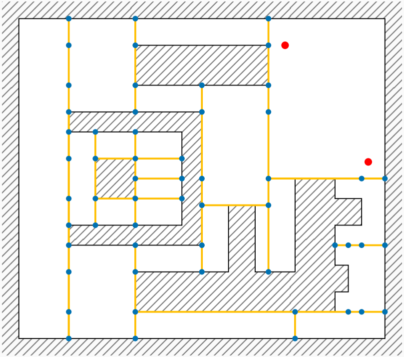
Routing Table Information	Bits
Region distance information	$\mathcal{O}(\log n)$
Region routing tables	$\mathcal{O}(\log n)$
Landmark graph	$\mathcal{O}(h^2 \cdot \log n)$
Total	$\mathcal{O}(h^2 \cdot \log n)$

Contributions - Routing Phase

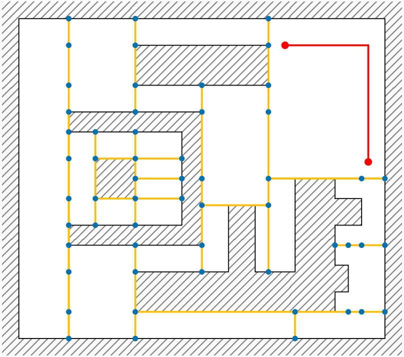


Contributions - Routing Phase

- If in same region as target:

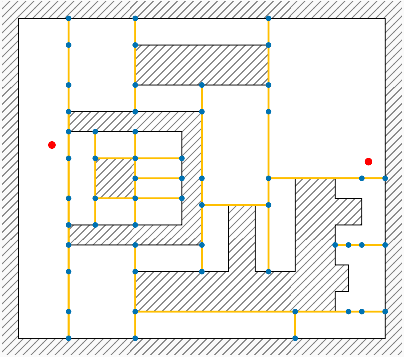


- If in same region as target:
    - Region routing tables



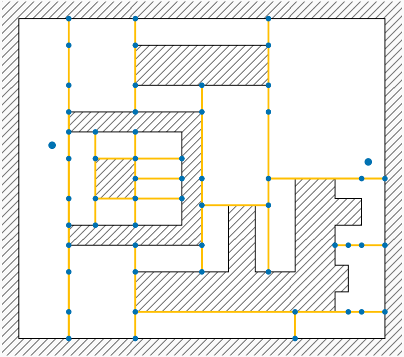
Contributions - Routing Phase

- If in same region as target:
  - Region routing tables
- Else:



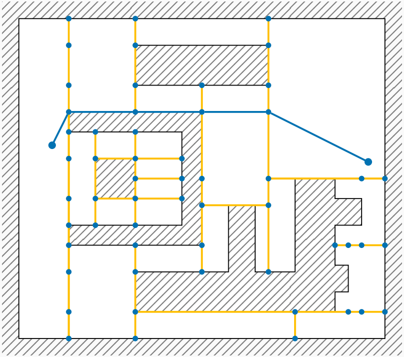
Contributions - Routing Phase

- If in same region as target:
  - Region routing tables
- Else:
  - Augment landmark graph



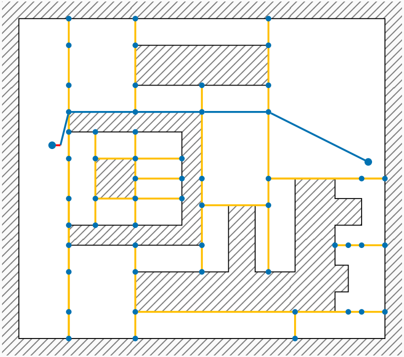
Contributions - Routing Phase

- If in same region as target:
  - Region routing tables
- Else:
  - Augment landmark graph
  - Locally solve SSSP



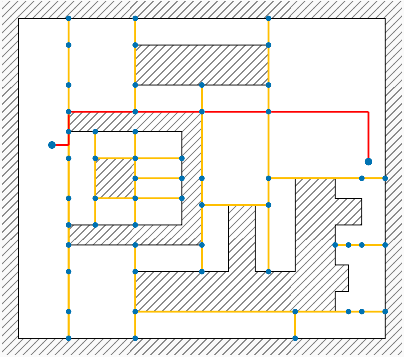
Contributions - Routing Phase

- If in same region as target:
  - Region routing tables
- Else:
  - Augment landmark graph
  - Locally solve SSSP
  - Forward to neighbor with smallest distance to next gate



Contributions - Routing Phase

- If in same region as target:
    - Region routing tables
- Else:
    - Augment landmark graph
    - Locally solve SSSP
    - Forward to neighbor with smallest distance to next gate



Contributions - Conclusion

- Exact for grid graphs, constant stretch in UDGs[7]
- Lower bound for preprocessing in general graphs: $\widetilde{\Omega}(n^{1/3})$[8]
- Upper bound for grid graphs: $O(h^2 + \log n)$

[7] Coy et al., *Near-Shortest Path Routing in Hybrid Communication Networks*.
[8] Kuhn and Schneider, "Routing Schemes and Distance Oracles in the Hybrid Model".

Contributions - Conclusion

- Exact for grid graphs, constant stretch in UDGs[7]
- Lower bound for preprocessing in general graphs: $\widetilde{\Omega}(n^{1/3})$[8]
- Upper bound for grid graphs: $O(h^2 + \log n)$

Future Work:

[7] Coy et al., *Near-Shortest Path Routing in Hybrid Communication Networks*.
[8] Kuhn and Schneider, "Routing Schemes and Distance Oracles in the Hybrid Model".

Contributions - Conclusion

- Exact for grid graphs, constant stretch in UDGs[7]
- Lower bound for preprocessing in general graphs: $\widetilde{\Omega}(n^{1/3})$[8]
- Upper bound for grid graphs: $O(h^2 + \log n)$

Future Work:

- Reduce h^2 to h for similar approach

[7] Coy et al., *Near-Shortest Path Routing in Hybrid Communication Networks*.
[8] Kuhn and Schneider, "Routing Schemes and Distance Oracles in the Hybrid Model".

Contributions - Conclusion

- Exact for grid graphs, constant stretch in UDGs[7]
- Lower bound for preprocessing in general graphs: $\widetilde{\Omega}(n^{1/3})$[8]
- Upper bound for grid graphs: $O(h^2 + \log n)$

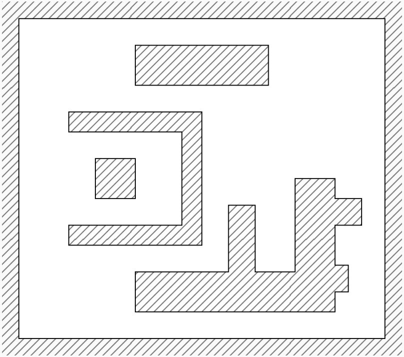
Future Work:

- Reduce h^2 to h for similar approach
- Different approaches without falling back to no holes

[7] Coy et al., *Near-Shortest Path Routing in Hybrid Communication Networks*.
[8] Kuhn and Schneider, "Routing Schemes and Distance Oracles in the Hybrid Model".

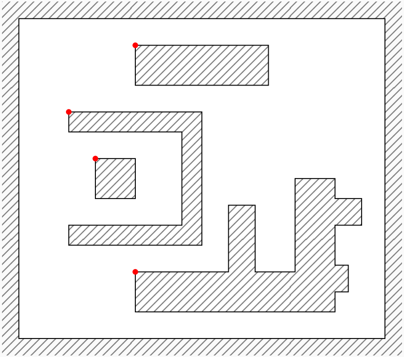
Thank you!

Contributions - Regionalization I: Simple Regions



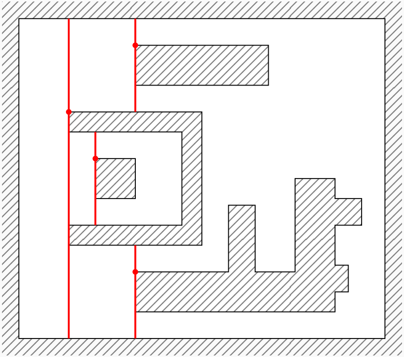
Contributions - Regionalization I: Simple Regions

- Mark leftmost node of each hole boundary



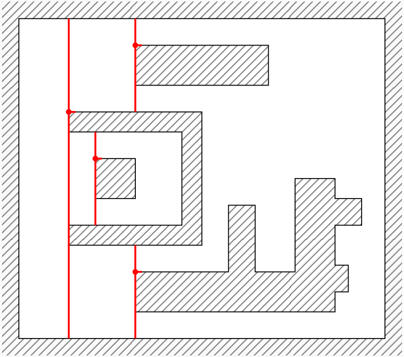
Contributions - Regionalization I: Simple Regions

- Mark leftmost node of each hole boundary
- Portals containing marked nodes are gates

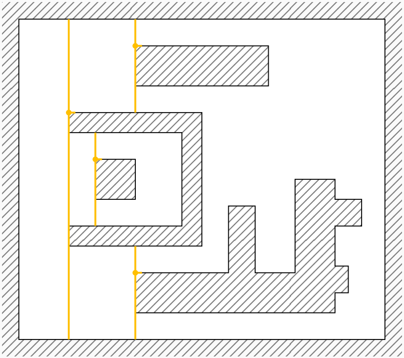


Contributions - Regionalization I: Simple Regions

- Mark leftmost node of each hole boundary
- Portals containing marked nodes are gates
- Marked nodes cut portals from hole's side

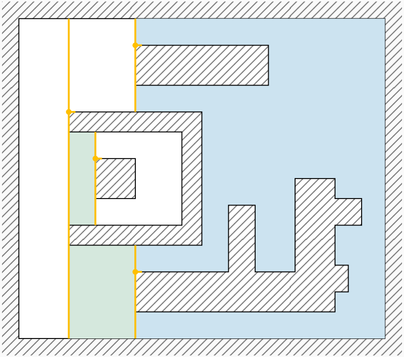


Contributions - Regionalization II: Tunnels



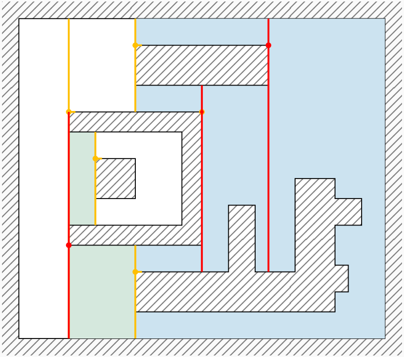
Contributions - Regionalization II: Tunnels

- Identify regions with > 2 gates



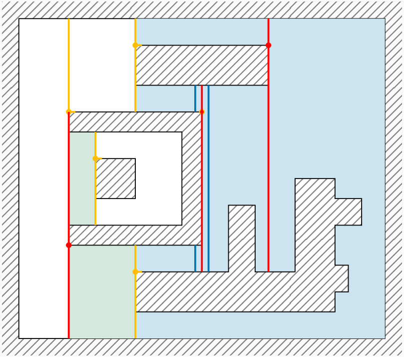
Contributions - Regionalization II: Tunnels

- Identify regions with > 2 gates
- Add portals splitting three gates as new gate

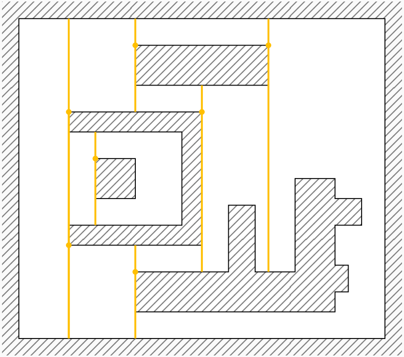


Contributions - Regionalization II: Tunnels

- Identify regions with > 2 gates
- Add portals splitting three gates as new gate
- Locally Checkable! Adjacent portals touch different holes

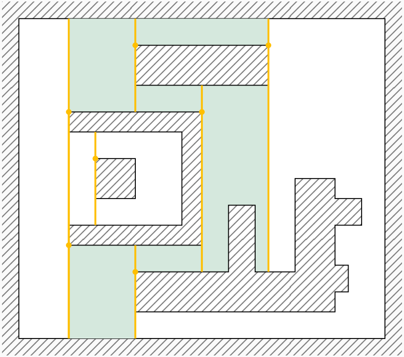


Contributions - Regionalization III: Path-Convex Regions



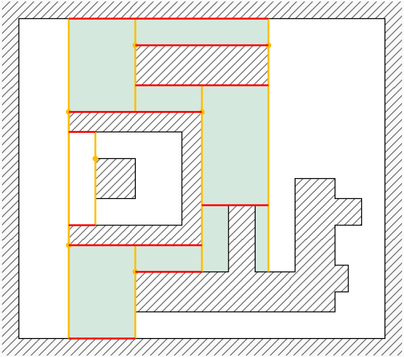
Contributions - Regionalization III: Path-Convex Regions

- Portals *see* each other



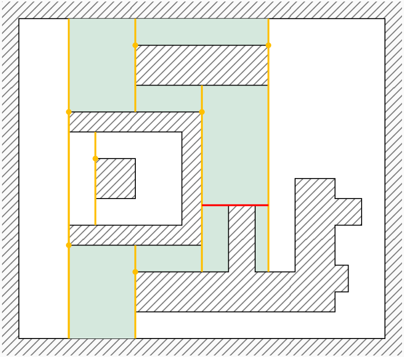
Contributions - Regionalization III: Path-Convex Regions

- Portals *see* each other
  - Bound Region with vertical distance 0



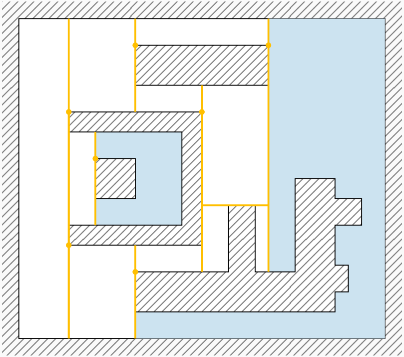
Contributions - Regionalization III: Path-Convex Regions

- Portals *see* each other
  - Bound Region with vertical distance 0



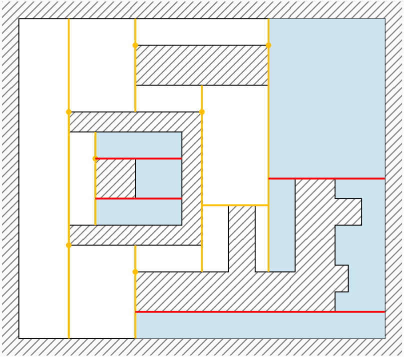
Contributions - Regionalization III: Path-Convex Regions

- Portals *see* each other
  - Bound Region with vertical distance 0
- Portals do not *see* each other



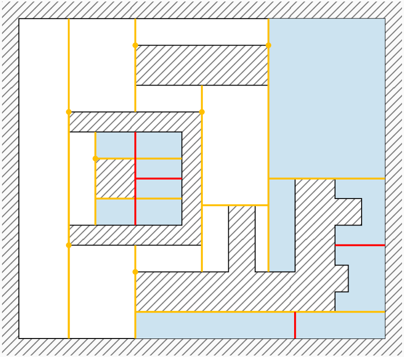
Contributions - Regionalization III: Path-Convex Regions

- Portals *see* each other
  - Bound Region with vertical distance 0
- Portals do not *see* each other
  - Add horizontal Gates through closest nodes

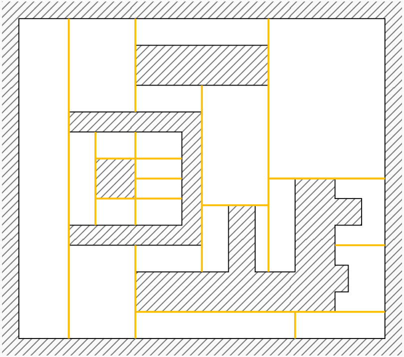


Contributions - Regionalization III: Path-Convex Regions

- Portals *see* each other
  - Bound Region with vertical distance 0
- Portals do not *see* each other
  - Add horizontal Gates through closest nodes
  - Add Gates at half horizontal & vertical distance

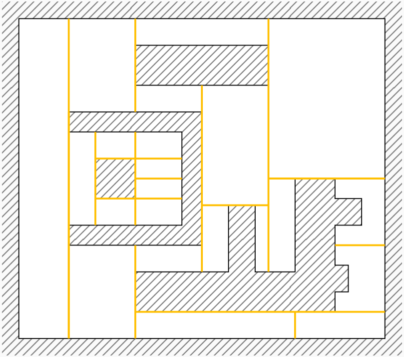


Contributions - Landmark Graph Nodes



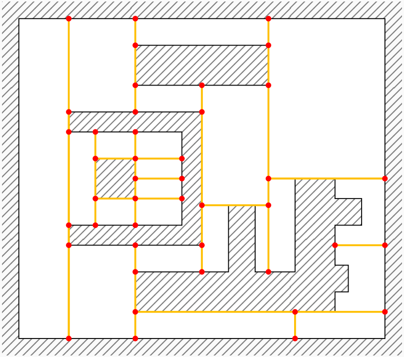
Contributions - Landmark Graph Nodes

- Mark key nodes as landmarks



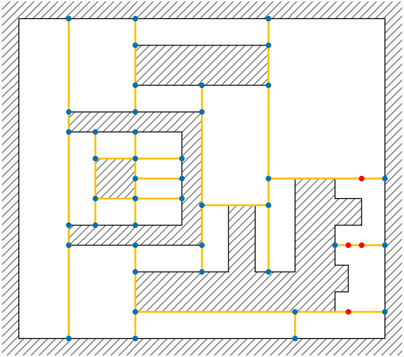
Contributions - Landmark Graph Nodes

- Mark key nodes as landmarks
  - Endpoints of Gates



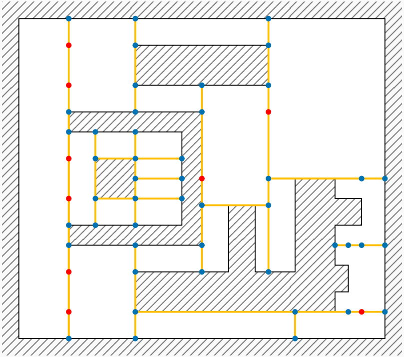
Contributions - Landmark Graph Nodes

- Mark key nodes as landmarks
  - Endpoints of Gates
  - Overhang Induced

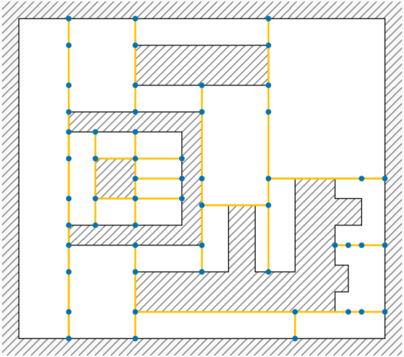


Contributions - Landmark Graph Nodes

- Mark key nodes as landmarks
  - Endpoints of Gates
  - Overhang Induced
  - Projections

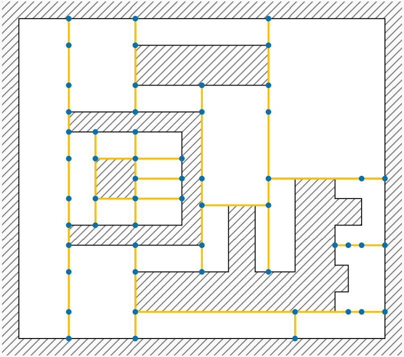


Contributions - Landmark Graph Edges



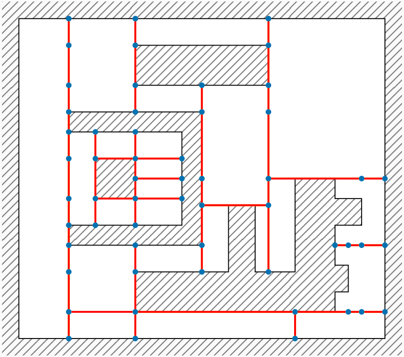
Contributions - Landmark Graph Edges

- Connect landmarks if



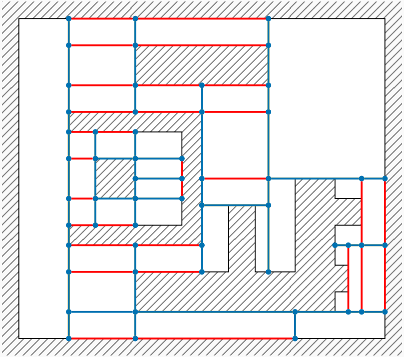
Contributions - Landmark Graph Edges

- Connect landmarks if
  - Adjacent on same Gate



Contributions - Landmark Graph Edges

- Connect landmarks if
  - Adjacent on same Gate
  - On adjacent Gates and closest



Contributions - Landmark Graph Edges

- Connect landmarks if
  - Adjacent on same Gate
  - On adjacent Gates and closest
- Add weights according to distances