Exploring Simple Grid Polygons

Christian Icking[1] **Tom Kamphans**[2]
Rolf Klein[2] Elmar Langetepe[2]

[1]University of Hagen, Praktische Informatik VI, Hagen, Germany.

[2]University of Bonn, Computer Science I, Bonn, Germany.

29th August 2005

- Robot, R, has to explore an unknown environment, P
- More precisely, find a tour that
  - visits every part of P at least once
  - returns to the robot's start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

The Problem

- Robot, R, has to explore an unknown environment, P
- More precisely, find a tour that
  - visits every part of P at least once
  - returns to the robot's start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

The Problem

- Robot, R, has to explore an unknown environment, P
- More precisely, find a tour that
  - visits every part of P at least once
  - returns to the robot's start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

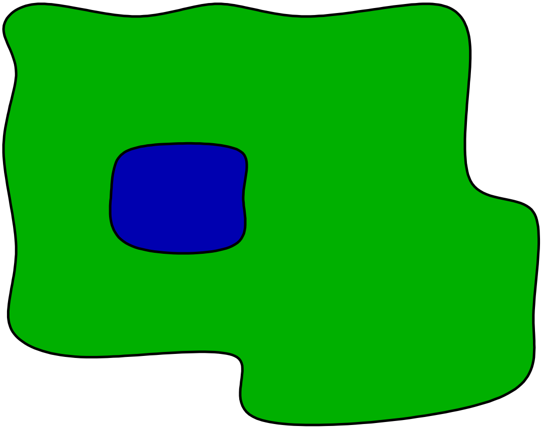
- Robot, R, has to explore an unknown environment, P
- More precisely, find a tour that
  - visits every part of P at least once
  - returns to the robot's start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

The Problem

- Robot, R, has to explore an unknown environment, P
- More precisely, find a tour that
  - visits every part of P at least once
  - returns to the robot's start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

The Problem

- Robot, R, has to explore an unknown environment, P
- More precisely, find a tour that
  - visits every part of P at least once
  - returns to the robot's start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

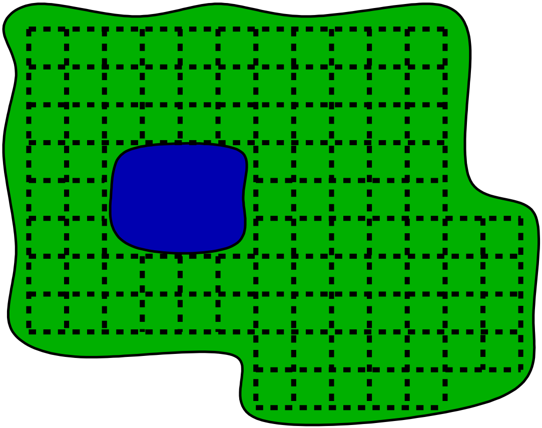


Grid polygon:

- Environment is subdivided by an integer grid
- Simple $\Rightarrow$ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell



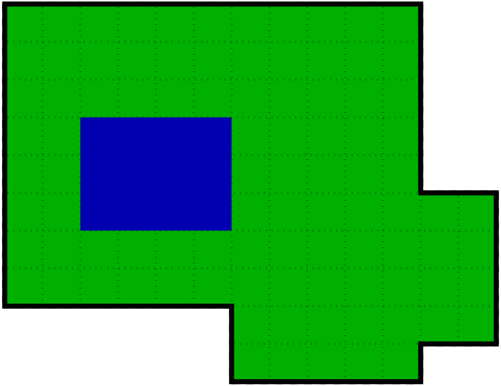
Grid polygon:

- Environment is subdivided by an integer grid
- Simple ⇒ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell

Environment and Robot

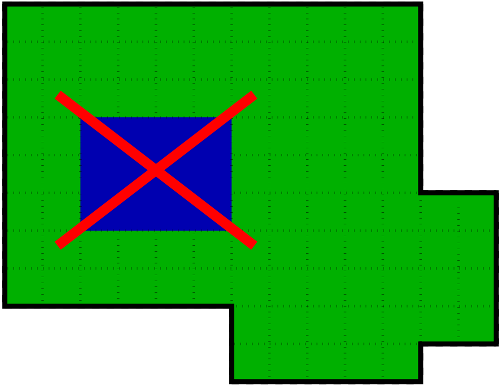


Grid polygon:

- Environment is subdivided by an integer grid
- Simple $\Rightarrow$ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell



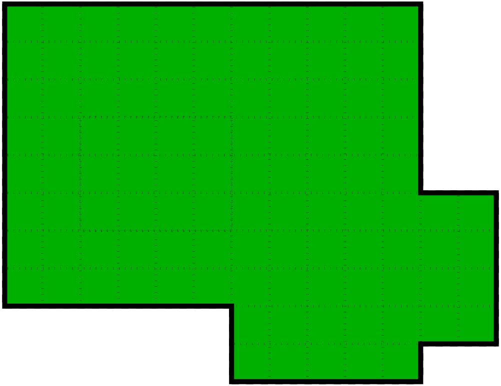
Grid polygon:

- Environment is subdivided by an integer grid
- Simple $\Rightarrow$ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell

Environment and Robot

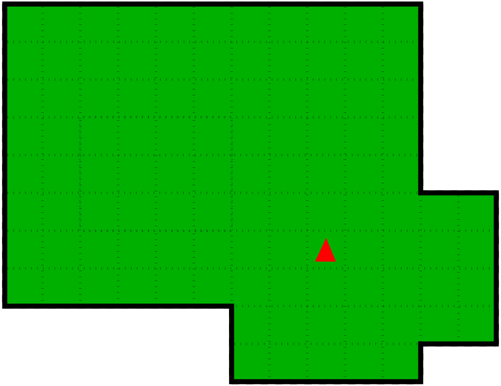


Grid polygon:

- Environment is subdivided by an integer grid
- Simple $\Rightarrow$ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell

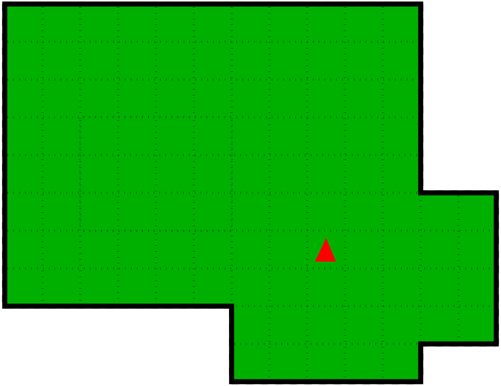


Grid polygon:

- Environment is subdivided by an integer grid
- Simple $\Rightarrow$ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell



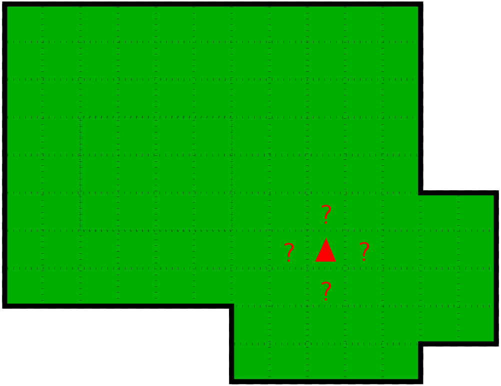
Grid polygon:

- Environment is subdivided by an integer grid
- Simple $\Rightarrow$ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell

Environment and Robot



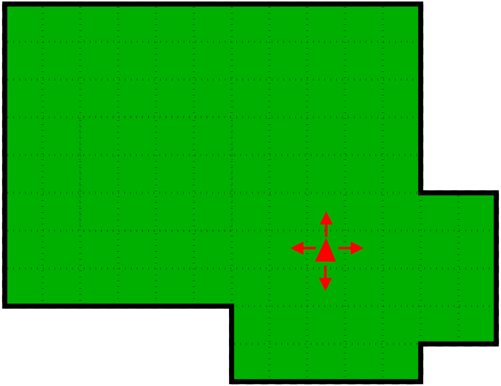
Grid polygon:

- Environment is subdivided by an integer grid
- Simple $\Rightarrow$ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell

Environment and Robot



Grid polygon:

- Environment is subdivided by an integer grid
- Simple $\Rightarrow$ No holes

Robot

- No vision
- Can sense 4 adjacent cells
- Can enter adjacent, *free* cell

Offline (i. e., environment is known to the robot)

- With holes:
  NP-hard [Itai, Papadimitriou, Szwarcfiter; 1982]
- Without holes:
  $\frac{4}{3}$-approximation [Ntafos; 1992]
  $\frac{6}{5}$-approximation [Arkin, Fekete, Mitchell; 2000]

Online

- With holes:
  [Icking, Kamphans, Klein, Langetepe; 2000]
  [Gabriely, Rimon; 2000]

Offline (i. e., environment is known to the robot)

- With holes:
  NP-hard [Itai, Papadimitriou, Szwarcfiter; 1982]
- Without holes:
  $\frac{4}{3}$-approximation [Ntafos; 1992]
  $\frac{6}{5}$-approximation [Arkin, Fekete, Mitchell; 2000]

Online

- With holes:
  [Icking, Kamphans, Klein, Langetepe; 2000]
  [Gabriely, Rimon; 2000]

Why Simple Polygons?

Theorem (IKKL; 2000)

Lower bound on the online exploration of grid polygons with *holes: 2.*

Theorem

There is a $\frac{4}{3}$-competitive online exploration strategy for polygons without *holes.*

Why Simple Polygons?

Theorem (IKKL; 2000)

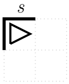
Lower bound on the online exploration of grid polygons with *holes: 2*.

Theorem

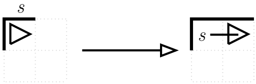
There is a $\frac{4}{3}$-competitive online exploration strategy for polygons without *holes.*

Theorem

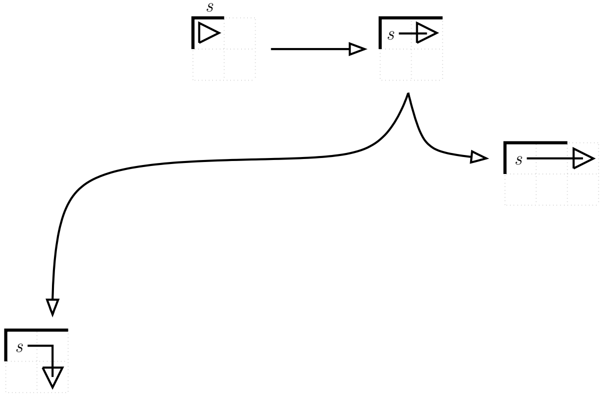
No online exploration strategy achieves a factor better than $\frac{7}{6}$ in simple grid polygon.


s

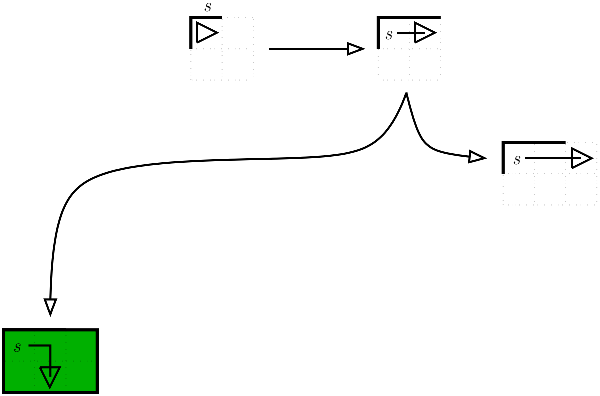
w. l. o. g.: East



South or East

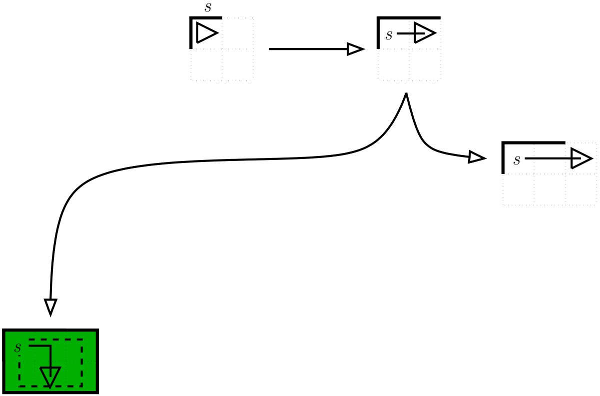


Close Polygon



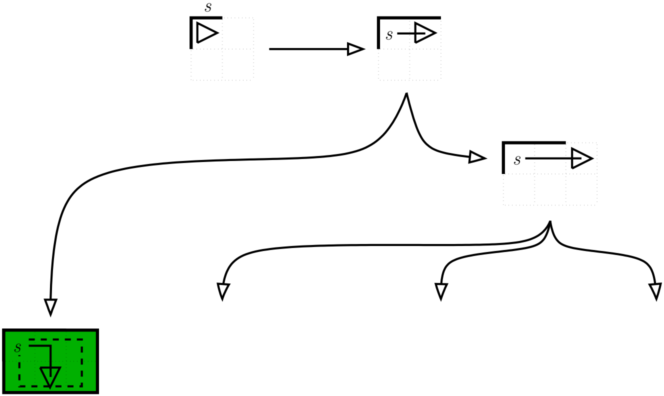
Proof: Lower Bound

Online vs. Optimal



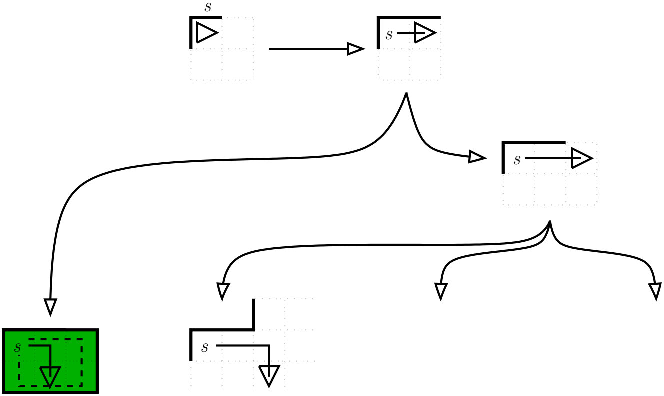
8/6

3 Possibilities:



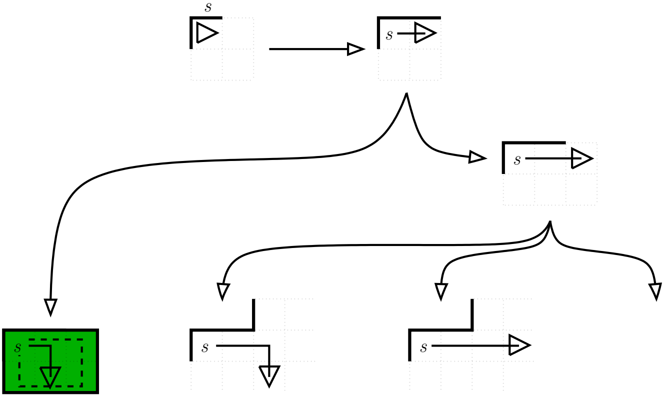
8/6

3 Possibilities: South,



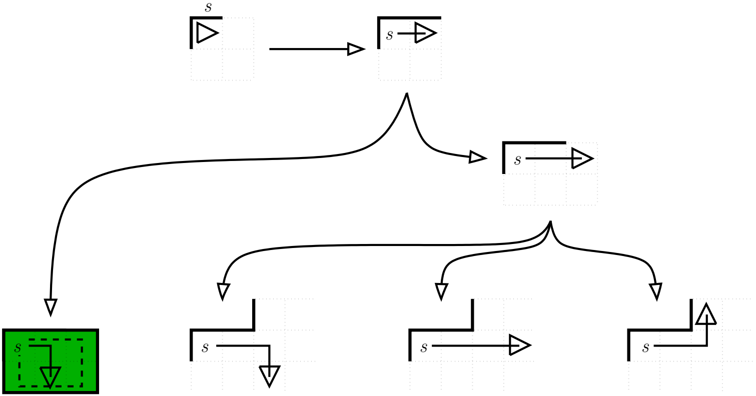
8/6

3 Possibilities: South, East,



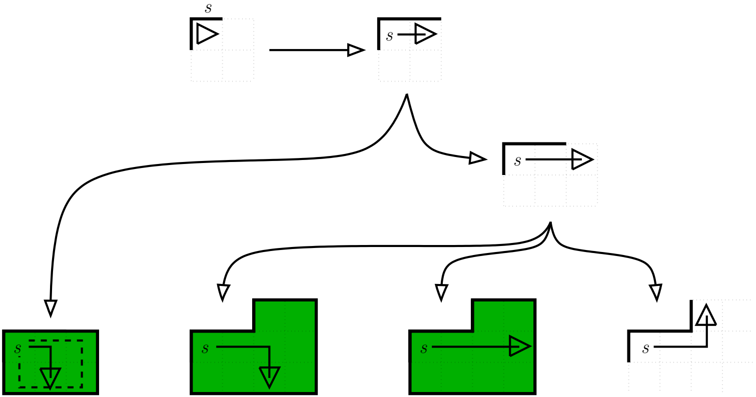
8/6

3 Possibilities: South, East, North



8/6

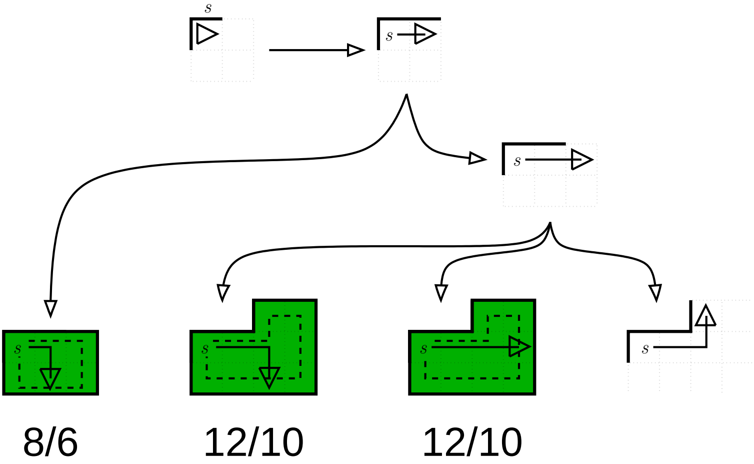
Close Polygon



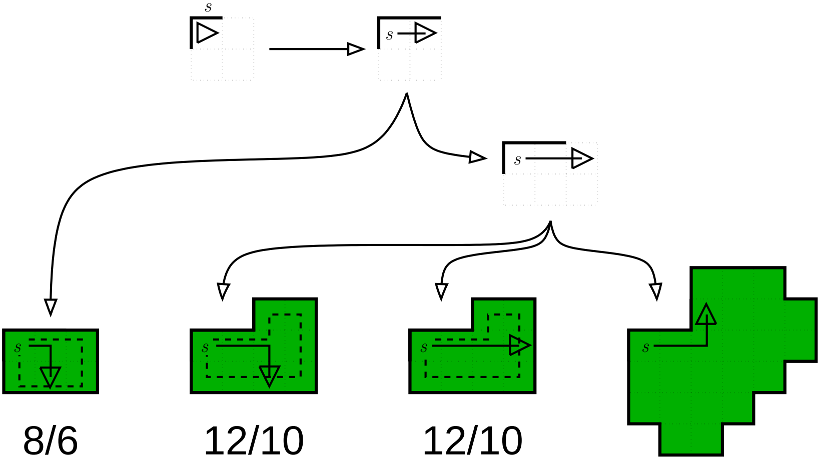
8/6

Proof: Lower Bound

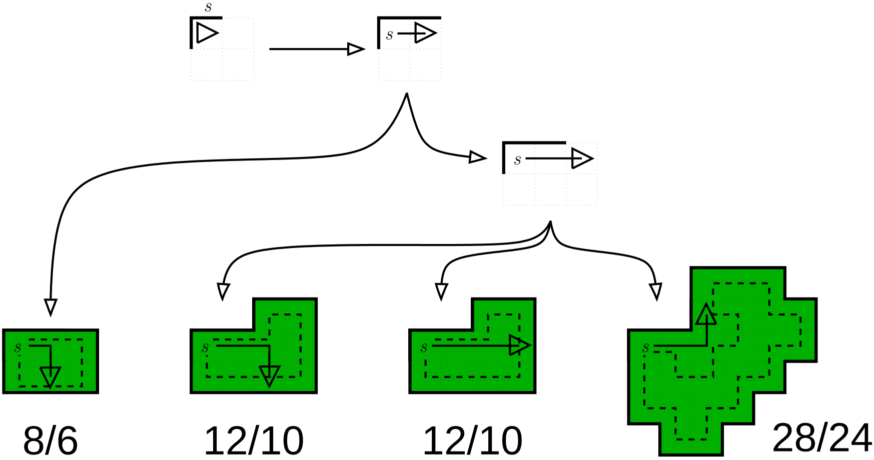
Online vs. Optimal



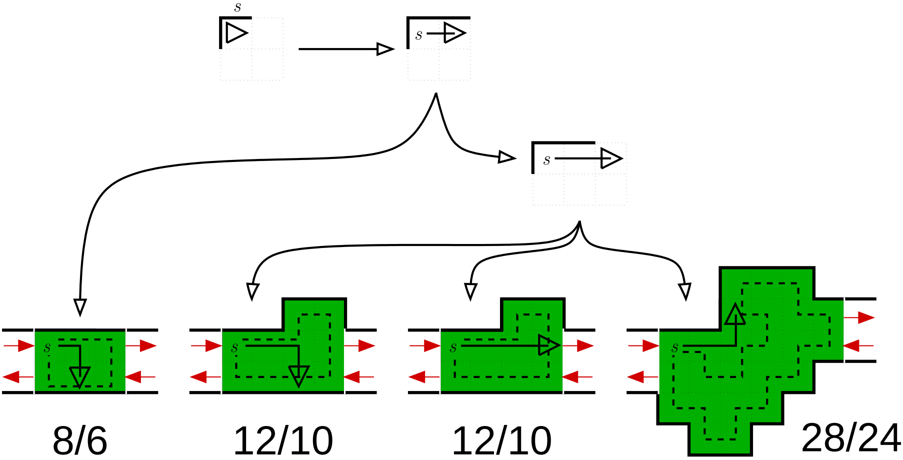
Close Polygon



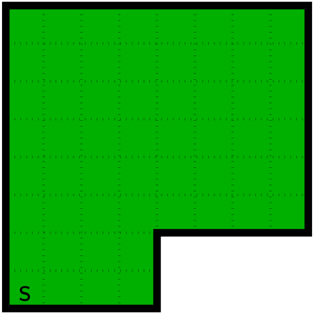
Online vs. Optimal



Polygons of arbitrary size

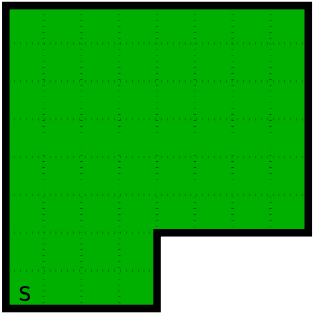


SmartDFS: An exploration strategy (1)



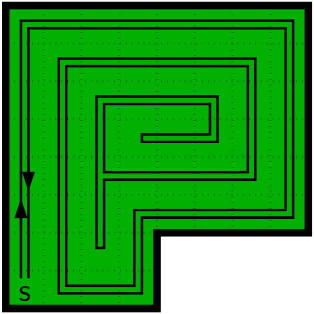
- First idea: Apply depth-first search (DFS)
- *Left-hand rule*: prefer step to the left over a straight step over a step to the right
- Visits *each* cell twice!

SmartDFS: An exploration strategy (1)



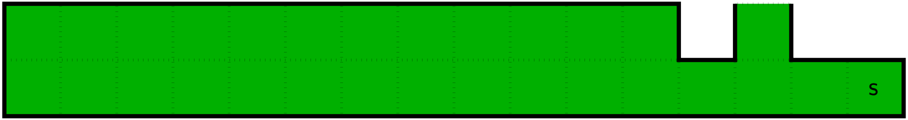
- First idea: Apply depth-first search (DFS)
- *Left-hand rule*: prefer step to the left over a straight step over a step to the right
- Visits *each* cell twice!

SmartDFS: An exploration strategy (1)



- First idea: Apply depth-first search (DFS)
- *Left-hand rule*: prefer step to the left over a straight step over a step to the right
- Visits *each* cell twice!

SmartDFS: An exploration strategy (2)

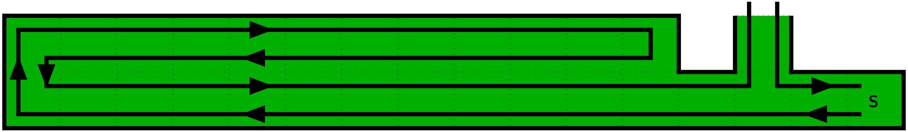


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (2)

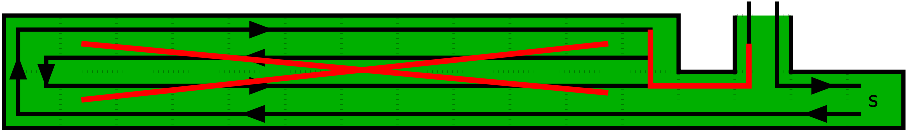


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (2)

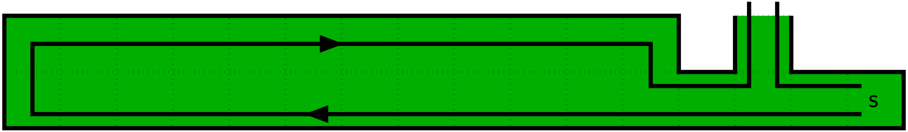


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (2)

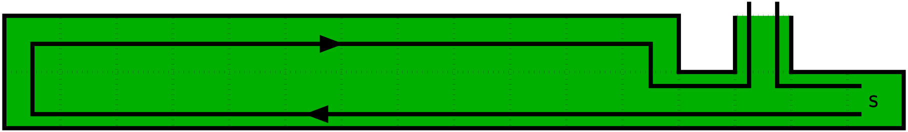


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (2)

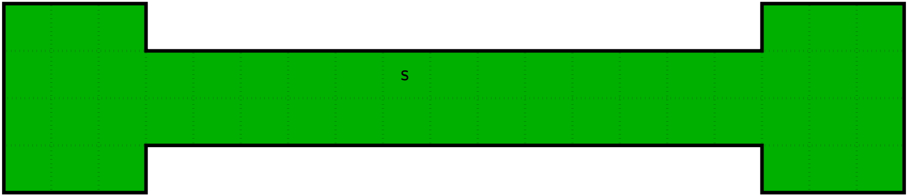


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (3)

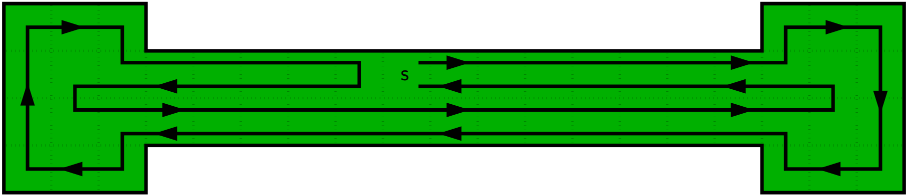


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to s
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of P farther away from the start).

SmartDFS: An exploration strategy (3)

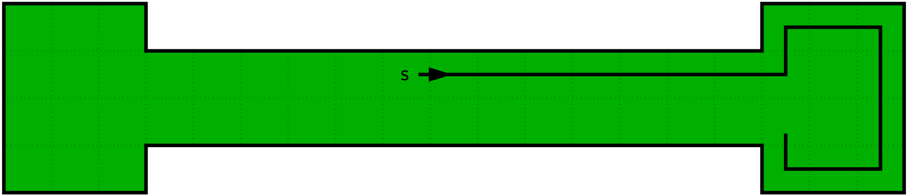


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to s
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of P farther away from the start).

SmartDFS: An exploration strategy (3)

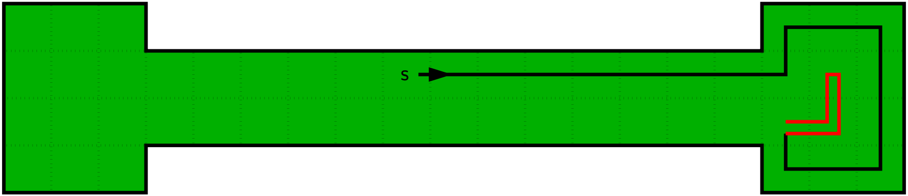


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of *P* farther away from the start).

SmartDFS: An exploration strategy (3)

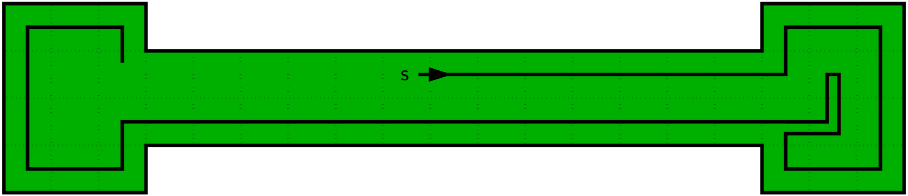


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to s
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of P farther away from the start).

SmartDFS: An exploration strategy (3)

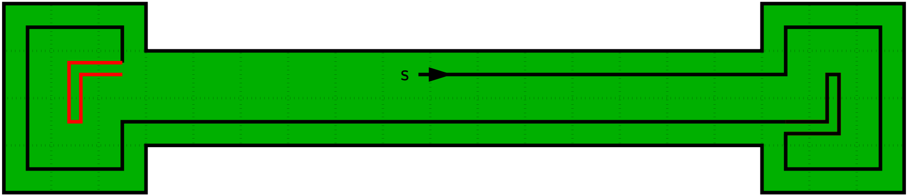


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of *P* farther away from the start).

SmartDFS: An exploration strategy (3)

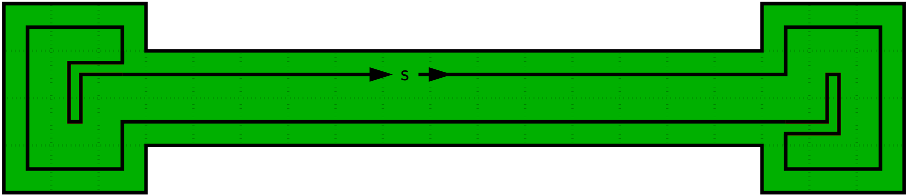


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of *P* farther away from the start).

SmartDFS: An exploration strategy (3)

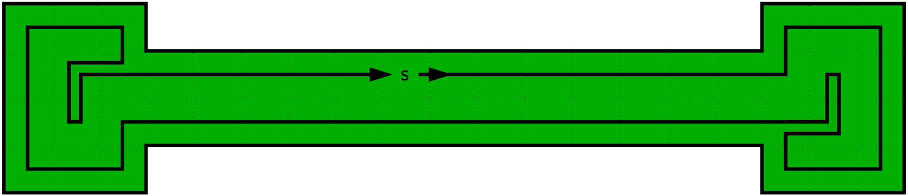


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to s
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of P farther away from the start).

SmartDFS: An exploration strategy (3)

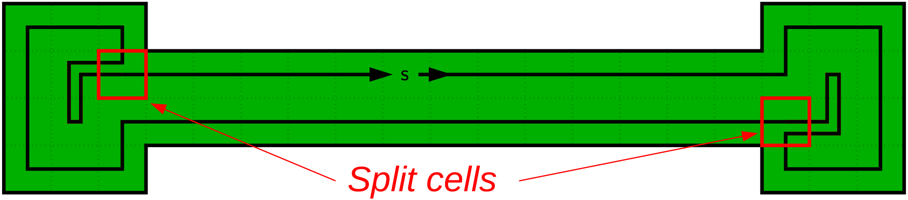


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to s
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of P farther away from the start).

SmartDFS: An exploration strategy (3)

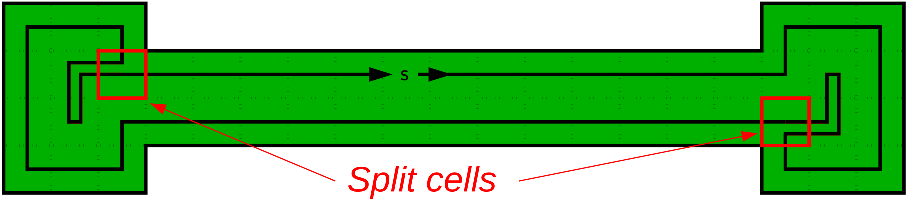


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of *P* farther away from the start).

SmartDFS: An exploration strategy (3)

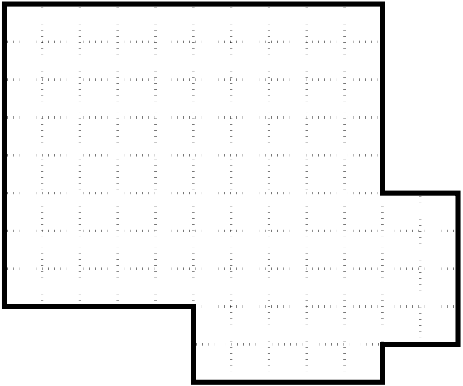


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to s
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

Detect and handle split cells (i. e., prefer parts of P farther away from the start).

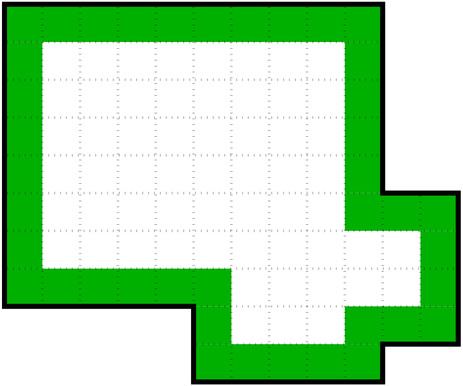
`http://www.geometrylab.de/Gridrobot/`



- *First layer* :=
  Boundary cells of P
- *1-offset* :=
  P without first layer
- Analogously: *Second layer*
- *2-offset*
- and so on
- E: #edges between free and blocked cells

Lemma (Number of edges)

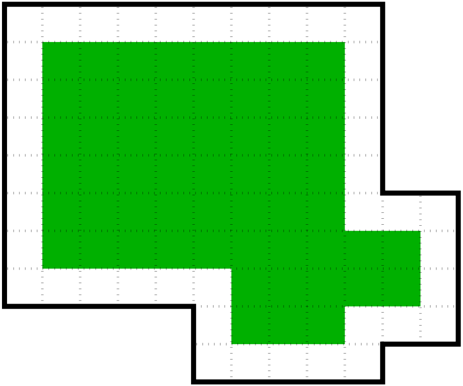
P' *is* ℓ*-offset of* $P \Rightarrow E(P') \leq E(P) - 8\ell$.



- *First layer* := Boundary cells of P
- *1-offset* := P without first layer
- Analogously: *Second layer*
- *2-offset*
- and so on
- E: #edges between free and blocked cells

Lemma (Number of edges)

P' is ℓ-offset of $P \Rightarrow E(P') \leq E(P) - 8\ell$.

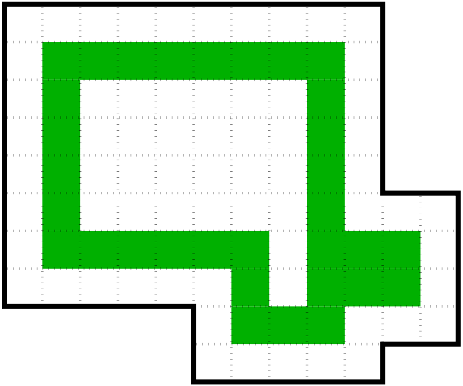


- *First layer* $:=$ Boundary cells of P
- *1-offset* $:=$ P without first layer
- Analogously: *Second layer*
- *2-offset*
- and so on
- E: #edges between free and blocked cells

Lemma (Number of edges)

P' is ℓ-offset of P $\Rightarrow E(P') \leq E(P) - 8\ell$.

Layer and Offset

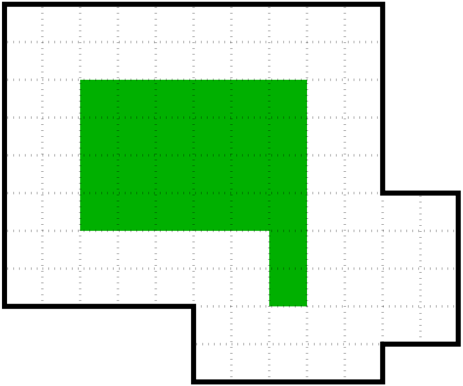


- *First layer* $:=$
  Boundary cells of P
- *1-offset* $:=$
  P without first layer
- Analogously: *Second layer*
- *2-offset*
- and so on
- E: #edges between free and blocked cells

Lemma (Number of edges)

P' *is* ℓ*-offset of* $P \Rightarrow E(P') \leq E(P) - 8\ell$.

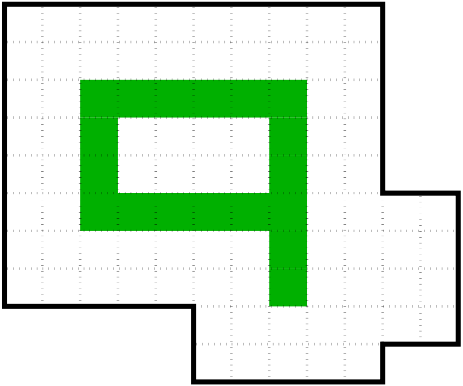
Layer and Offset



- *First layer* $:=$ Boundary cells of P
- *1-offset* $:=$ P without first layer
- Analogously: *Second layer*
- *2-offset*
- and so on
- E: #edges between free and blocked cells

Lemma (Number of edges)

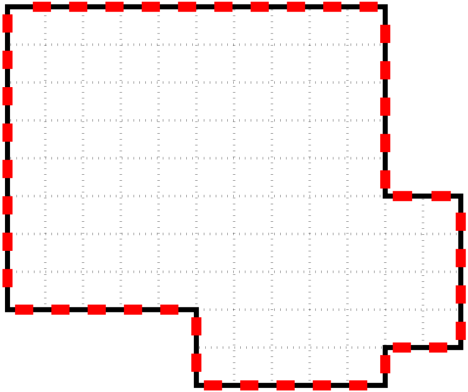
P' is ℓ-offset of P $\Rightarrow E(P') \leq E(P) - 8\ell$.



- *First layer* $:=$ Boundary cells of P
- *1-offset* $:=$ P without first layer
- Analogously: *Second layer*
- *2-offset*
- and so on
- E: #edges between free and blocked cells

Lemma (Number of edges)

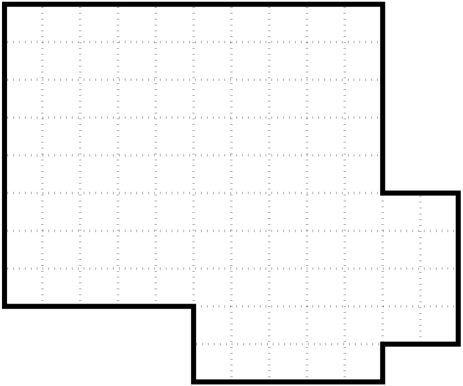
P' is ℓ-offset of P $\Rightarrow E(P') \leq E(P) - 8\ell$.



- *First layer* $:=$
  Boundary cells of P
- *1-offset* $:=$
  P without first layer
- Analogously: *Second layer*
- *2-offset*
- and so on
- E: #edges between free and blocked cells

Lemma (Number of edges)

P' *is* ℓ*-offset of* $P \Rightarrow E(P') \leq E(P) - 8\ell$.



- *First layer* $:=$ Boundary cells of P
- *1-offset* $:=$ P without first layer
- Analogously: *Second layer*
- *2-offset*
- and so on
- E: #edges between free and blocked cells

Lemma (Number of edges)

P' is ℓ-offset of P $\Rightarrow E(P') \leq E(P) - 8\ell$.

Lemma (Shortest Path)

Shortest path between two cells in $P \leq \frac{1}{2}E(P) - 2$.

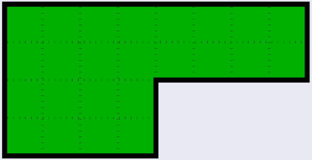
Proof sketch.

- Worst case:
  Both cells in the first layer
- $|\pi_{cw}| = |\pi_{ccw}|$
  $= \frac{1}{2} \cdot$ #cells in the first layer
- #cells in the first layer
  $\leq$ #edges $- 4$

Lemma (Shortest Path)

Shortest path between two cells in $P \le \frac{1}{2}E(P) - 2$.

Proof sketch.



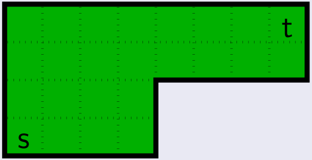
- Worst case:
  Both cells in the first layer
- $|\pi_{\mathrm{cw}}| = |\pi_{\mathrm{ccw}}|$
  $= \frac{1}{2} \cdot$ #cells in the first layer
- #cells in the first layer
  $\le$ #edges $- 4$

$\square$

Lemma (Shortest Path)

Shortest path between two cells in $P \leq \frac{1}{2}E(P) - 2$.

Proof sketch.



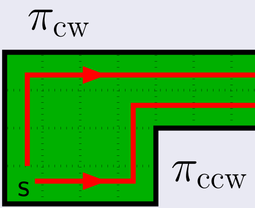
- Worst case:
  Both cells in the first layer
- $|\pi_{\mathrm{cw}}| = |\pi_{\mathrm{ccw}}|$
  $= \frac{1}{2} \cdot$ #cells in the first layer
- #cells in the first layer
  $\leq$ #edges $- 4$

$\square$

Lemma (Shortest Path)

Shortest path between two cells in $P \leq \frac{1}{2}E(P) - 2$.

Proof sketch.



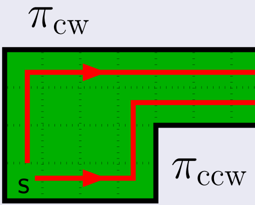
- Worst case:
  Both cells in the first layer
- $|\pi_{\mathrm{cw}}| = |\pi_{\mathrm{ccw}}|$
  $= \frac{1}{2} \cdot$ #cells in the first layer
- #cells in the first layer
  $\leq$ #edges $- 4$

□

Lemma (Shortest Path)

Shortest path between two cells in $P \leq \frac{1}{2}E(P) - 2$.

Proof sketch.



- Worst case:
  Both cells in the first layer
- $|\pi_{\mathrm{cw}}| = |\pi_{\mathrm{ccw}}|$
  $= \frac{1}{2} \cdot$ #cells in the first layer
- #cells in the first layer
  $\leq$ #edges $- 4$

$\square$

Upper for the Number of Steps (1)

Theorem (Number of Steps)

$$S \leq C + \frac{1}{2}E - 3 \qquad \textit{(tight!)}$$

(S: #Steps from cell to cell, C: #Cells, E: #Boundary edges)

Proof sketch

- Induction on the number of split cells
- Induction base: No split cell
  - Visit every cell in $C - 1$ steps
  - Return to s in $\leq \frac{1}{2}E - 2$ steps (Lemma)

Theorem (Number of Steps)

$$S \leq C + \frac{1}{2}E - 3 \qquad \textit{(tight!)}$$

(S: #Steps from cell to cell, C: #Cells, E: #Boundary edges)

Proof sketch

- Induction on the number of split cells
- Induction base: No split cell
-     Visit every cell in $C - 1$ steps
-     Return to s in $\leq \frac{1}{2}E - 2$ steps (Lemma)

Upper for the Number of Steps (1)

Theorem (Number of Steps)

$$S \leq C + \frac{1}{2}E - 3 \qquad \textit{(tight!)}$$

(S: #Steps from cell to cell, C: #Cells, E: #Boundary edges)

Proof sketch

- Induction on the number of split cells
- Induction base: No split cell
  - Visit every cell in $C - 1$ steps
  - Return to s in $\leq \frac{1}{2}E - 2$ steps (Lemma)

Theorem (Number of Steps)

$$S \leq C + \frac{1}{2}E - 3 \qquad \textit{(tight!)}$$

(S: #Steps from cell to cell, C: #Cells, E: #Boundary edges)

Proof sketch

- Induction on the number of split cells
- Induction base: No split cell
- Visit every cell in $C - 1$ steps
- Return to s in $\leq \frac{1}{2}E - 2$ steps (Lemma)

Upper for the Number of Steps (1)

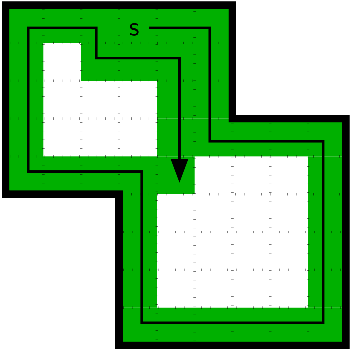
Theorem (Number of Steps)

$$S \leq C + \frac{1}{2}E - 3 \qquad \textit{(tight!)}$$

(S: #Steps from cell to cell, C: #Cells, E: #Boundary edges)

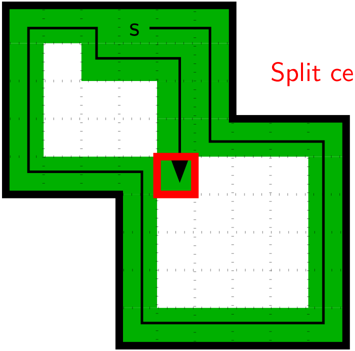
Proof sketch

- Induction on the number of split cells
- Induction base: No split cell
- Visit every cell in $C - 1$ steps
- Return to s in $\leq \frac{1}{2}E - 2$ steps (Lemma)



Induction step:

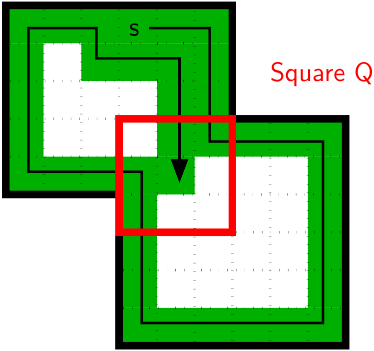
- Explore up to the first split cell
- Define square Q around split cell
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_1 is unexplored part of P_1
- K_2 is unexplored part of P_2
- Path in $P_2 \backslash K_2$ visits no cell twice



Induction step:

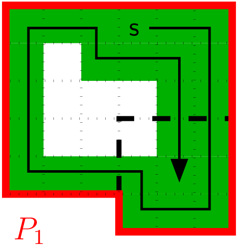
- Explore up to the first split cell
- Define square Q around split cell
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_1 is unexplored part of P_1
- K_2 is unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice

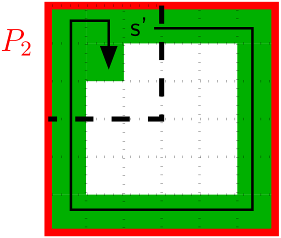
Proof Sketch (2)



Induction step:

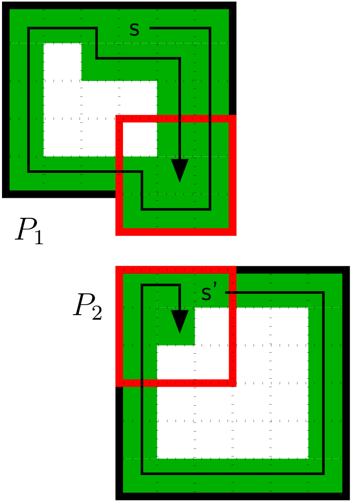
- Explore up to the first split cell
- Define square Q around split cell
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_1 is unexplored part of P_1
- K_2 is unexplored part of P_2
- Path in $P_2 \backslash K_2$ visits no cell twice





Induction step:

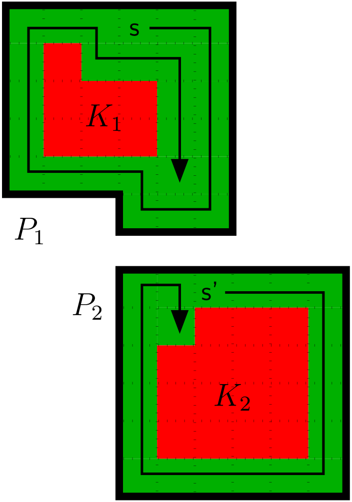
- Explore up to the first split cell
- Define square Q around split cell
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_1 is unexplored part of P_1
- K_2 is unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice



Induction step:

- Explore up to the first split cell
- Define square Q around split cell
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_1 is unexplored part of P_1
- K_2 is unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice

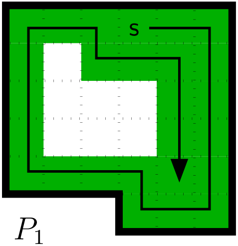
Proof Sketch (2)

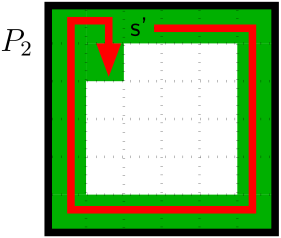


Induction step:

- Explore up to the first split cell
- Define square Q around split cell
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_1 is unexplored part of P_1
- K_2 is unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice

Proof Sketch (2)





Induction step:

- Explore up to the first split cell
- Define square Q around split cell
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_1 is unexplored part of P_1
- K_2 is unexplored part of P_2
- Path in $P_2 \backslash K_2$ visits no cell twice

- Path in $P_2 \backslash K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and K_2
- K_2 is offset of P_2; bound edges in K_2 by edges in P_2
- Charge edges of P_1 and P_2 to edges of P and Q
- Edges in Q: $E(Q) = 4(2q - 1)$

$$\Rightarrow \quad S(P) \leq C + \frac{1}{2} E(P) - 3$$

- Path in $P_2 \backslash K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and K_2
- K_2 is offset of P_2; bound edges in K_2 by edges in P_2
- Charge edges of P_1 and P_2 to edges of P and Q
- Edges in Q: $E(Q) = 4(2q - 1)$

$$\Rightarrow \quad S(P) \leq C + \frac{1}{2} E(P) - 3$$

Proof Sketch (3)

- Path in $P_2 \backslash K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and K_2
- K_2 is offset of P_2; bound edges in K_2 by edges in P_2
- Charge edges of P_1 and P_2 to edges of P and Q
- Edges in Q: $E(Q) = 4(2q - 1)$

$$\Rightarrow \quad S(P) \leq C + \frac{1}{2} E(P) - 3$$

Proof Sketch (3)

- Path in $P_2 \backslash K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and K_2
- K_2 is offset of P_2; bound edges in K_2 by edges in P_2
- Charge edges of P_1 and P_2 to edges of P and Q
- Edges in Q: $E(Q) = 4(2q - 1)$

$$\Rightarrow \quad S(P) \leq C + \frac{1}{2} E(P) - 3$$

Proof Sketch (3)

- Path in $P_2 \backslash K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and K_2
- K_2 is offset of P_2; bound edges in K_2 by edges in P_2
- Charge edges of P_1 and P_2 to edges of P and Q
- Edges in Q: $E(Q) = 4(2q - 1)$

$$\Rightarrow \quad S(P) \leq C + \frac{1}{2} E(P) - 3$$

Proof Sketch (3)

$$\text{excess}(P) \leq \text{excess}(P_1) + \text{excess}(P_2)$$

- Path in $P_2 \backslash K_2$ visits no cell twice

$$\text{excess}(P) \leq \text{excess}(P_1) + \text{excess}(K_2 \cup \{c\}) + 1$$

- Apply induction hypothesis to P_1 and K_2

$$\text{excess}(P) \leq \tfrac{1}{2}E(P_1) - 3 + \tfrac{1}{2}E(K_2 \cup \{c\}) - 3 + 1$$

- K_2 is offset of P_2; bound edges in K_2 by edges in P_2

$$\text{excess}(P) \leq \tfrac{1}{2}E(P_1) + \tfrac{1}{2}E(P_2) - 8q - 5$$

- Charge edges of P_1 and P_2 to edges of P and Q

$$\text{excess}(P) \leq \tfrac{1}{2}E(P) + \tfrac{1}{2}E(Q) - 8q - 5$$

- Edges in Q: $E(Q) = 4(2q - 1)$

$$\text{excess}(P) \leq \tfrac{1}{2}E(P) - 3$$

$$\Rightarrow \quad S(P) \leq C + \tfrac{1}{2}E(P) - 3$$

Theorem (Competitivity)

SmartDFS is $\frac{4}{3}$ competitive (i. e., $S_{\text{SmartDFS}} \leq \frac{4}{3}\, S_{\text{Optimal}}$)

Definition

Narrow passage: Corridors of width 1 or 2.

Definition

Uncritical polygon: neither narrow passages nor split cells in the first layer.

Theorem (Competitivity)

SmartDFS is $\frac{4}{3}$ competitive (i. e., $S_{\text{SmartDFS}} \leq \frac{4}{3}\, S_{\text{Optimal}}$)

Definition

Narrow passage: Corridors of width 1 or 2.

Definition

Uncritical polygon: neither narrow passages nor split cells in the first layer.

Theorem (Competitivity)

SmartDFS is $\frac{4}{3}$ *competitive (i. e.,* $S_{\text{SmartDFS}} \leq \frac{4}{3} S_{\text{Optimal}}$*)*

Definition

Narrow passage: Corridors of width 1 or 2.

Definition

Uncritical polygon: neither narrow passages nor split cells in the first layer.

Competitivity (2)

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.

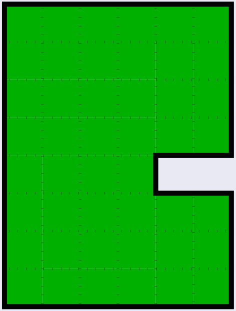
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

□

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



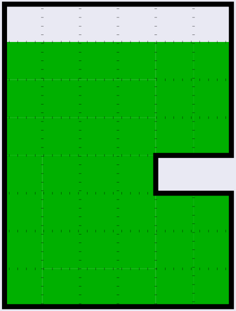
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

□

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



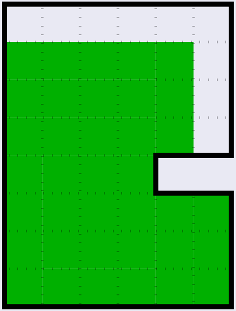
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

$\square$

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

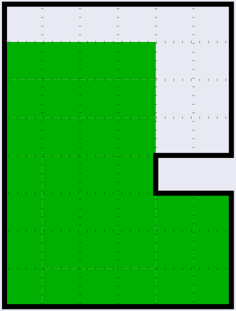
□

Competitivity (2)

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



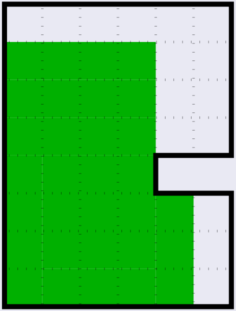
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

□

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



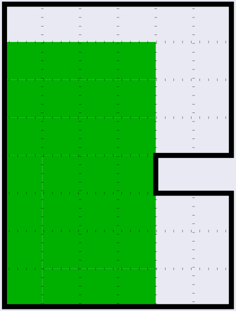
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

□

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

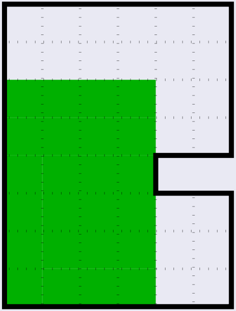
$\square$

Competitivity (2)

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



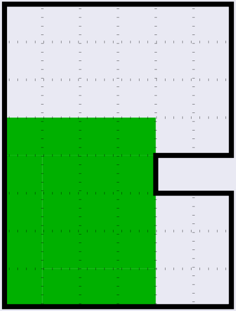
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

□

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



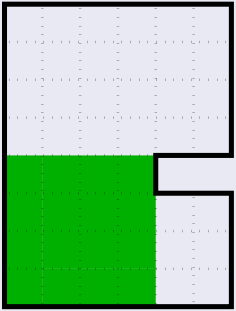
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

$\square$

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



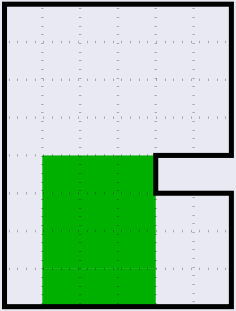
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

□

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



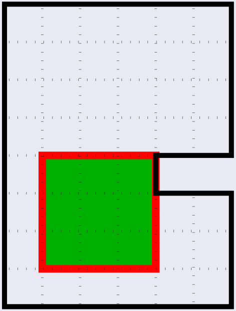
- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

□

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.

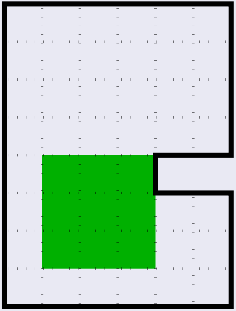


- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

Lemma (Edges in uncritical polygons)

For uncritical grid polygons: $E(P) \leq \frac{2}{3}C(P) + 6$

Proof.



- Successively remove row or column of at least 3 cells, maintaining the uncritical property
- Ends with 3×3 polygon, $E = \frac{2}{3}C + 6$
- $E \leq \frac{2}{3}C + 6$ fulfilled in every step

□

Competitivity (3)

Lemma (Exploration of uncritical polygons)

For uncritical grid polygons: $S(P) \leq C(P) + \frac{1}{2}E(P) - 5$.

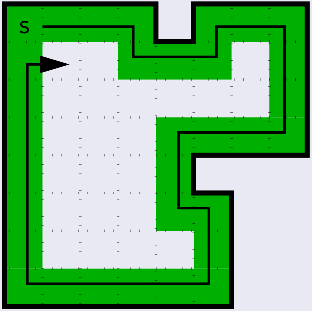
Proof sketch

- $S(P) \leq C(P) + \frac{1}{2}E(P) - 3$ shown
- Used shortest path lemma ($sp(c, s) \leq \frac{1}{2}E(P) - 2$)
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Lemma (Exploration of uncritical polygons)

For uncritical grid polygons: $S(P) \leq C(P) + \frac{1}{2}E(P) - 5$.

Proof sketch



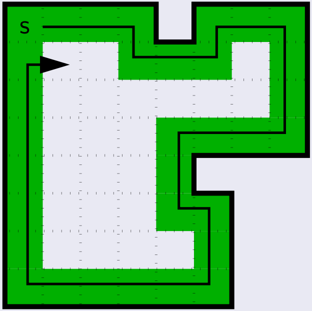
- $S(P) \leq C(P) + \frac{1}{2}E(P) - 3$ shown
- Used shortest path lemma ($sp(c, s) \leq \frac{1}{2}E(P) - 2$)
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Competitivity (3)

Lemma (Exploration of uncritical polygons)

For uncritical grid polygons: $S(P) \leq C(P) + \frac{1}{2}E(P) - 5$.

Proof sketch

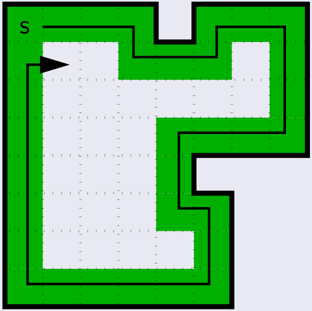


- $S(P) \leq C(P) + \frac{1}{2}E(P) - 3$ shown
- Used shortest path lemma ($sp(c, s) \leq \frac{1}{2}E(P) - 2$)
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Lemma (Exploration of uncritical polygons)

For uncritical grid polygons: $S(P) \leq C(P) + \frac{1}{2}E(P) - 5$.

Proof sketch



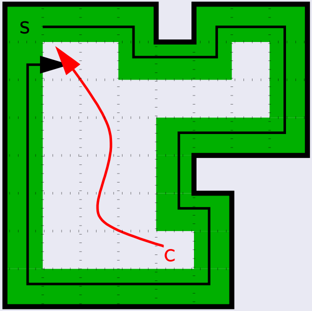
- $S(P) \leq C(P) + \frac{1}{2}E(P) - 3$ shown
- Used shortest path lemma ($sp(c,s) \leq \frac{1}{2}E(P) - 2$)
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Competitivity (3)

Lemma (Exploration of uncritical polygons)

For uncritical grid polygons: $S(P) \leq C(P) + \frac{1}{2}E(P) - 5$.

Proof sketch



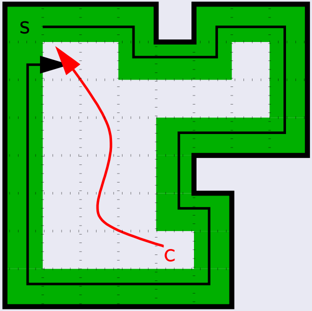
- $S(P) \leq C(P) + \frac{1}{2}E(P) - 3$ shown
- Used shortest path lemma ($sp(c,s) \leq \frac{1}{2}E(P) - 2$)
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Competitivity (3)

Lemma (Exploration of uncritical polygons)

For uncritical grid polygons: $S(P) \leq C(P) + \frac{1}{2}E(P) - 5$.

Proof sketch



- $S(P) \leq C(P) + \frac{1}{2}E(P) - 3$ shown
- Used shortest path lemma ($sp(c, s) \leq \frac{1}{2}E(P) - 2$)
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Theorem (Competitivity)

SmartDFS is $\frac{4}{3}$ competitive.

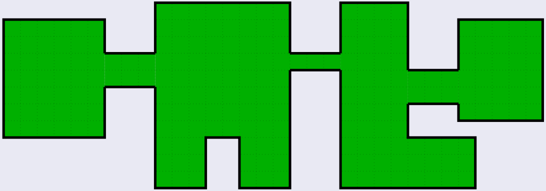
Proof

- Remove narrow passages (explored optimally)
- $\Rightarrow$ Split P into P_i
- Consider P_i separately

Theorem (Competitivity)

SmartDFS is $\frac{4}{3}$ competitive.

Proof

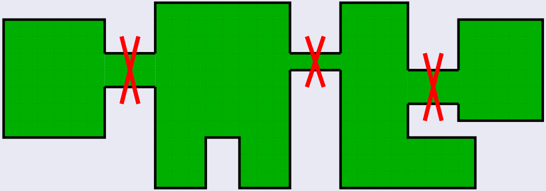


- Remove narrow passages (explored optimally)
- $\Rightarrow$ Split P into P_i
- Consider P_i separately

Theorem (Competitivity)

SmartDFS is $\frac{4}{3}$ competitive.

Proof

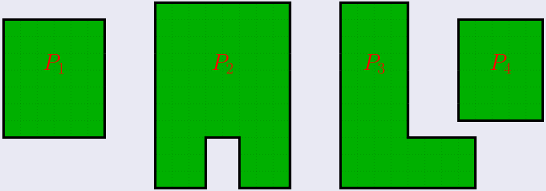


- Remove narrow passages (explored optimally)
- $\Rightarrow$ Split P into P_i
- Consider P_i separately

Theorem (Competitivity)

SmartDFS is $\frac{4}{3}$ competitive.

Proof

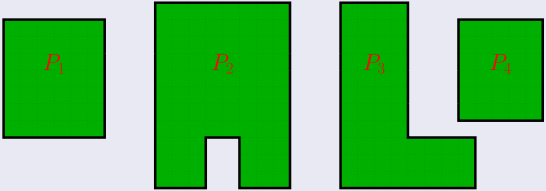


- Remove narrow passages (explored optimally)
- $\Rightarrow$ Split P into P_i
- Consider P_i separately

Theorem (Competitivity)

SmartDFS is $\frac{4}{3}$ competitive.

Proof



- Remove narrow passages (explored optimally)
- $\Rightarrow$ Split P into P_i
- Consider P_i separately

- Show $S(P_i) \leq \frac{4}{3}C(P_i) - 2$
  by induction on the number of split cells in the first layer
- Ind. base: No split cell $\Rightarrow$ uncritical polygon $\Rightarrow$

$$
\begin{aligned}
S(P_i) &\leq C(P_i) + \frac{1}{2}E(P_i) - 5 && \text{by exploration lemma} \\
&\leq C(P_i) + \frac{1}{2}\left(\frac{2}{3}C(P_i) + 6\right) - 5 && \text{by edges lemma} \\
&= \frac{4}{3}C(P_i) - 2
\end{aligned}
$$

Competitivity Proof (2)

- Show $S(P_i) \leq \frac{4}{3}C(P_i) - 2$ by induction on the number of split cells in the first layer
- Ind. base: No split cell $\Rightarrow$ uncritical polygon $\Rightarrow$

$$\begin{aligned} S(P_i) &\leq C(P_i) + \frac{1}{2}E(P_i) - 5 && \text{by exploration lemma} \\ &\leq C(P_i) + \frac{1}{2}\left(\frac{2}{3}C(P_i) + 6\right) - 5 && \text{by edges lemma} \\ &= \frac{4}{3}C(P_i) - 2 \end{aligned}$$

- Show $S(P_i) \leq \frac{4}{3}C(P_i) - 2$
  by induction on the number of split cells in the first layer
- Ind. base: No split cell $\Rightarrow$ uncritical polygon $\Rightarrow$

$$
\begin{aligned}
S(P_i) &\leq C(P_i) + \frac{1}{2}E(P_i) - 5 && \text{by exploration lemma} \\
&\leq C(P_i) + \frac{1}{2}\left(\frac{2}{3}C(P_i) + 6\right) - 5 && \text{by edges lemma} \\
&= \frac{4}{3}C(P_i) - 2
\end{aligned}
$$

- Show $S(P_i) \leq \frac{4}{3}C(P_i) - 2$
  by induction on the number of split cells in the first layer
- Ind. base: No split cell $\Rightarrow$ uncritical polygon $\Rightarrow$

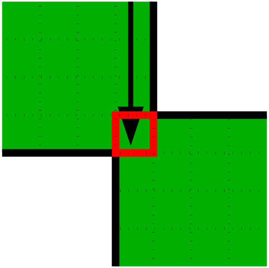
$$\begin{aligned} S(P_i) &\leq C(P_i) + \frac{1}{2}E(P_i) - 5 && \text{by exploration lemma} \\ &\leq C(P_i) + \frac{1}{2}\left(\frac{2}{3}C(P_i) + 6\right) - 5 && \text{by edges lemma} \\ &= \frac{4}{3}C(P_i) - 2 \end{aligned}$$

Competitivity Proof (2)

- Show $S(P_i) \leq \frac{4}{3}C(P_i) - 2$
  by induction on the number of split cells in the first layer
- Ind. base: No split cell $\Rightarrow$ uncritical polygon $\Rightarrow$

$$\begin{aligned}
S(P_i) &\leq C(P_i) + \frac{1}{2}E(P_i) - 5 && \text{by exploration lemma} \\
&\leq C(P_i) + \frac{1}{2}\left(\frac{2}{3}C(P_i) + 6\right) - 5 && \text{by edges lemma} \\
&= \frac{4}{3}C(P_i) - 2
\end{aligned}$$

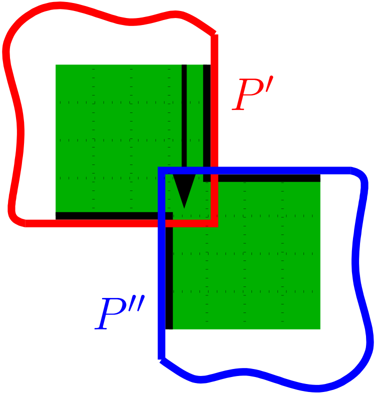
Ind. step, case 1: New component was never visited before



- Split P_i into P', P''
- $S(P_i) = S(P') + S(P'')$
- $C(P_i) = C(P') + C(P'') - 1$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') \\ &\leq \frac{4}{3}C(P') - 2 + \frac{4}{3}C(P'') - 2 \\ &= \frac{4}{3}C(P_i) + \frac{4}{3} - 4 \\ &< \frac{4}{3}C(P_i) - 2 \end{aligned}$$

Ind. step, case 1: New component was never visited before

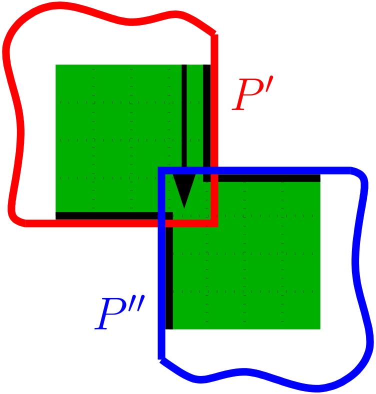


- Split P_i into P', P''
- $S(P_i) = S(P') + S(P'')$
- $C(P_i) = C(P') + C(P'') - 1$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') \\ &\leq \frac{4}{3}C(P') - 2 + \frac{4}{3}C(P'') - 2 \\ &= \frac{4}{3}C(P_i) + \frac{4}{3} - 4 \\ &< \frac{4}{3}C(P_i) - 2 \end{aligned}$$

Competitivity Proof (3)

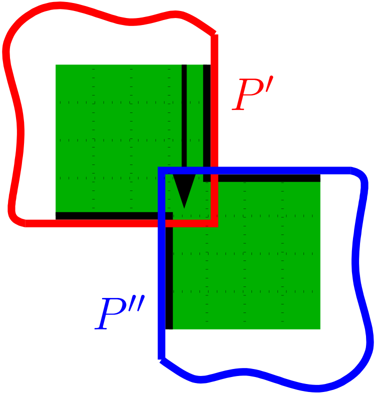
Ind. step, case 1: New component was never visited before



- Split P_i into P', P''
- $S(P_i) = S(P') + S(P'')$
- $C(P_i) = C(P') + C(P'') - 1$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') \\ &\leq \frac{4}{3}C(P') - 2 + \frac{4}{3}C(P'') - 2 \\ &= \frac{4}{3}C(P_i) + \frac{4}{3} - 4 \\ &< \frac{4}{3}C(P_i) - 2 \end{aligned}$$

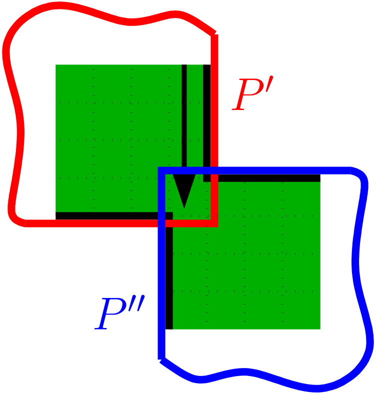
Ind. step, case 1: New component was never visited before



- Split P_i into P', P''
- $S(P_i) = S(P') + S(P'')$
- $C(P_i) = C(P') + C(P'') - 1$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') \\ &\leq \frac{4}{3}C(P') - 2 + \frac{4}{3}C(P'') - 2 \\ &= \frac{4}{3}C(P_i) + \frac{4}{3} - 4 \\ &< \frac{4}{3}C(P_i) - 2 \end{aligned}$$

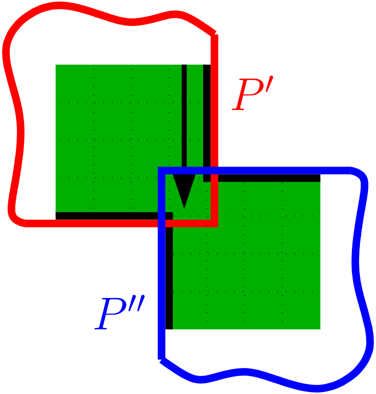
Ind. step, case 1: New component was never visited before



- Split P_i into P', P''
- $S(P_i) = S(P') + S(P'')$
- $C(P_i) = C(P') + C(P'') - 1$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') \\ &\leq \frac{4}{3}C(P') - 2 + \frac{4}{3}C(P'') - 2 \\ &= \frac{4}{3}C(P_i) + \frac{4}{3} - 4 \\ &< \frac{4}{3}C(P_i) - 2 \end{aligned}$$

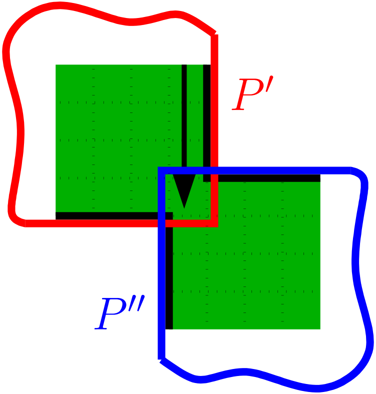
Ind. step, case 1: New component was never visited before



- Split P_i into P', P''
- $S(P_i) = S(P') + S(P'')$
- $C(P_i) = C(P') + C(P'') - 1$

$$
\begin{aligned}
S(P_i) &= S(P') + S(P'') \\
&\leq \frac{4}{3} C(P') - 2 + \frac{4}{3} C(P'') - 2 \\
&= \frac{4}{3} C(P_i) + \frac{4}{3} - 4 \\
&< \frac{4}{3} C(P_i) - 2
\end{aligned}
$$

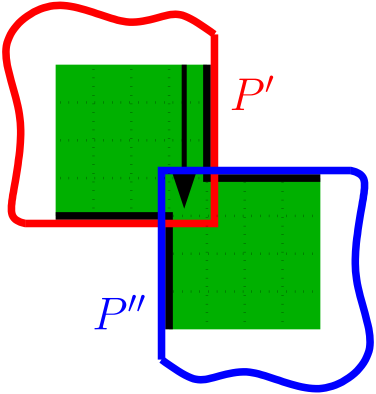
Ind. step, case 1: New component was never visited before



- Split P_i into P', P''
- $S(P_i) = S(P') + S(P'')$
- $C(P_i) = C(P') + C(P'') - 1$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') \\ &\leq \frac{4}{3}C(P') - 2 + \frac{4}{3}C(P'') - 2 \\ &= \frac{4}{3}C(P_i) + \frac{4}{3} - 4 \\ &< \frac{4}{3}C(P_i) - 2 \end{aligned}$$

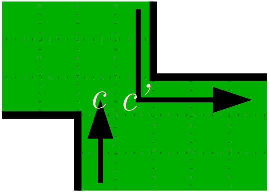
Ind. step, case 1: New component was never visited before

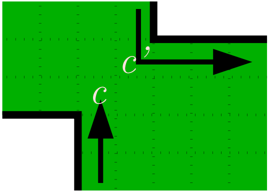


- Split P_i into P', P''
- $S(P_i) = S(P') + S(P'')$
- $C(P_i) = C(P') + C(P'') - 1$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') \\ &\leq \frac{4}{3} C(P') - 2 + \frac{4}{3} C(P'') - 2 \\ &= \frac{4}{3} C(P_i) + \frac{4}{3} - 4 \\ &< \frac{4}{3} C(P_i) - 2 \end{aligned}$$

Ind. step, case 2: Robot meets cell c' touching split cell c



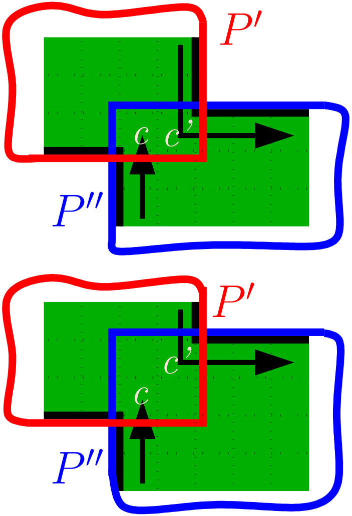


- Split P_i into P', P''
- $Q :=$ largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') - |Q| \\ &\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\ &= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\ &< \frac{4}{3}C(P_i) - 2 \qquad \square \end{aligned}$$

Competitivity Proof (4)

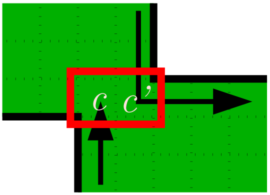
Ind. step, case 2: Robot meets cell c' touching split cell c

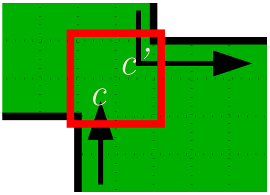


- Split P_i into P', P''
- Q := largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') - |Q| \\ &\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\ &= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\ &< \frac{4}{3}C(P_i) - 2 \qquad \square \end{aligned}$$

Ind. step, case 2: Robot meets cell c' touching split cell c



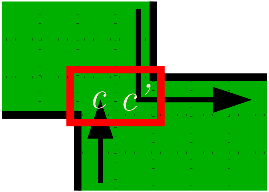


- Split P_i into P', P''
- $Q :=$ largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$

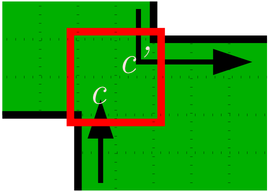
$$
\begin{aligned}
S(P_i) &= S(P') + S(P'') - |Q| \\
&\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\
&= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\
&< \frac{4}{3}C(P_i) - 2 \qquad \square
\end{aligned}
$$

Competitivity Proof (4)

Ind. step, case 2: Robot meets cell c' touching split cell c

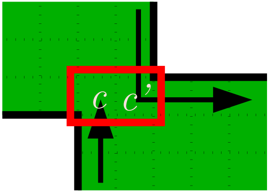


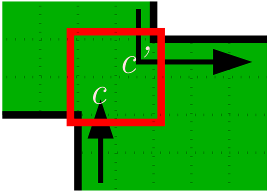
- Split P_i into P', P''
- $Q :=$ largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$



$$\begin{aligned} S(P_i) &= S(P') + S(P'') - |Q| \\ &\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\ &= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\ &< \frac{4}{3}C(P_i) - 2 \qquad \square \end{aligned}$$

Ind. step, case 2: Robot meets cell c' touching split cell c



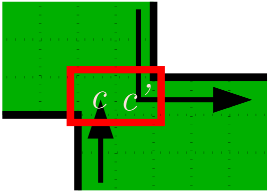


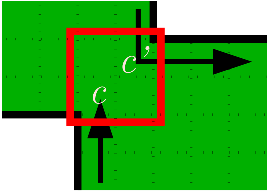
- Split P_i into P', P''
- $Q :=$ largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') - |Q| \\ &\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\ &= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\ &< \frac{4}{3}C(P_i) - 2 \qquad \square \end{aligned}$$

Competitivity Proof (4)

Ind. step, case 2: Robot meets cell c' touching split cell c

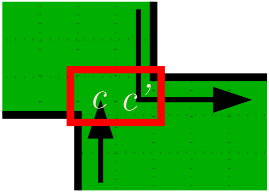


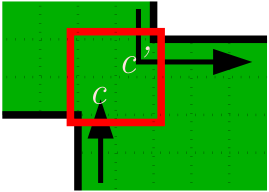


- Split P_i into P', P''
- $Q :=$ largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') - |Q| \\ &\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\ &= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\ &< \frac{4}{3}C(P_i) - 2 \qquad \square \end{aligned}$$

Ind. step, case 2: Robot meets cell c' touching split cell c

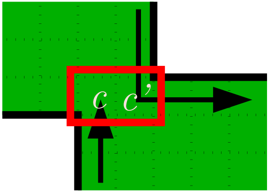


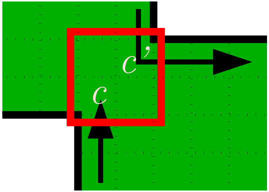


- Split P_i into P', P''
- $Q :=$ largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$

$$
\begin{aligned}
S(P_i) &= S(P') + S(P'') - |Q| \\
&\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\
&= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\
&< \frac{4}{3}C(P_i) - 2 \qquad \square
\end{aligned}
$$

Ind. step, case 2: Robot meets cell c' touching split cell c



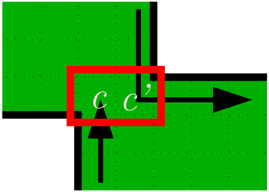


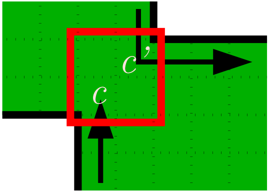
- Split P_i into P', P''
- $Q :=$ largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') - |Q| \\ &\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\ &= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\ &< \frac{4}{3}C(P_i) - 2 \qquad \square \end{aligned}$$

Competitivity Proof (4)

Ind. step, case 2: Robot meets cell c' touching split cell c





- Split P_i into P', P''
- $Q :=$ largest rectangle containing both c, c'
- $C(P_i) = C(P') + C(P'') - |Q|$

$$\begin{aligned} S(P_i) &= S(P') + S(P'') - |Q| \\ &\leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 4 - |Q| \\ &= \frac{4}{3}C(P_i) + \frac{1}{3}(|Q| - 6) - 2 \\ &< \frac{4}{3}C(P_i) - 2 \qquad \square \end{aligned}$$

Problem: Online exploration of simple grid polygons

- Lower Bound: $\frac{7}{6}$
- Exploration strategy *SmartDFS*
- $S \leq C + \frac{1}{2}E - 3$
- $\frac{4}{3}$-competitive

$\longrightarrow$ Accepted for COCOON 2005

- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower Bound: $\frac{7}{6}$
- Exploration strategy *SmartDFS*
- $S \leq C + \frac{1}{2}E - 3$
- $\frac{4}{3}$-competitive

$\longrightarrow$ Accepted for COCOON 2005

- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower Bound: $\frac{7}{6}$
- Exploration strategy *SmartDFS*
- $S \leq C + \frac{1}{2}E - 3$
- $\frac{4}{3}$-competitive

$\longrightarrow$ Accepted for COCOON 2005

- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower Bound: $\frac{7}{6}$
- Exploration strategy *SmartDFS*
- $S \leq C + \frac{1}{2}E - 3$
- $\frac{4}{3}$-competitive

$\longrightarrow$ Accepted for COCOON 2005

- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower Bound: $\frac{7}{6}$
- Exploration strategy *SmartDFS*
- $S \leq C + \frac{1}{2}E - 3$
- $\frac{4}{3}$-competitive

$\longrightarrow$ Accepted for COCOON 2005

- ToDo: Close the gap!

Summary

Problem: Online exploration of simple grid polygons

- Lower Bound: $\frac{7}{6}$
- Exploration strategy *SmartDFS*
- $S \leq C + \frac{1}{2}E - 3$
- $\frac{4}{3}$-competitive

$\longrightarrow$ Accepted for COCOON 2005

- ToDo: Close the gap!

Summary

Problem: Online exploration of simple grid polygons

- Lower Bound: $\frac{7}{6}$
- Exploration strategy *SmartDFS*
- $S \le C + \frac{1}{2}E - 3$
- $\frac{4}{3}$-competitive

$\longrightarrow$ Accepted for COCOON 2005

- ToDo: Close the gap!

Thank you!