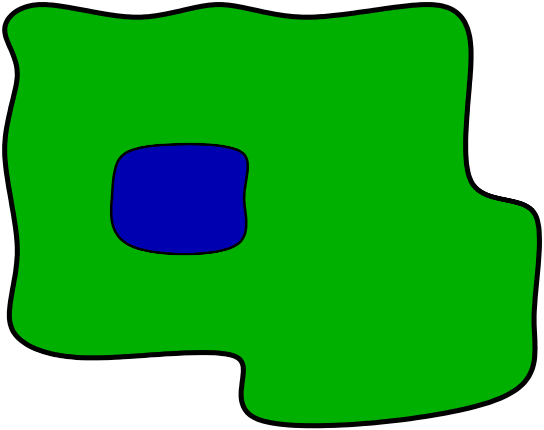
Exploring Simple Triangular and Hexagonal Grid Polygons Online

Daniel Herrmann[1] **Tom Kamphans**[2]
Elmar Langetepe[1]

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

[2]Braunschweig University of Technology, Computer Science, Algorithms Group, Braunschweig, Germany.

19.3.2008

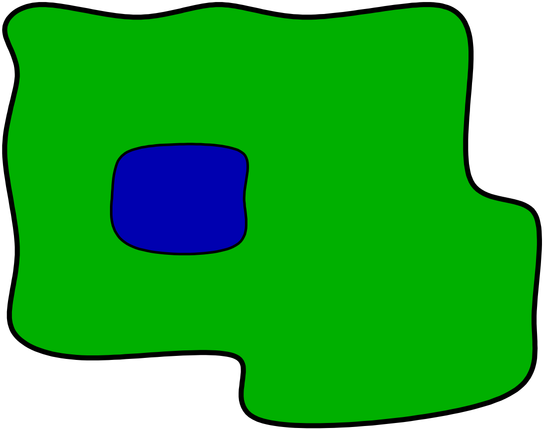


Environment

- Convenient for motion planning tasks:

  Subdivide env. by integer grid
- E.g.: cell size $\approx$ size of robot's tool
- Simple $\Leftrightarrow$ No holes

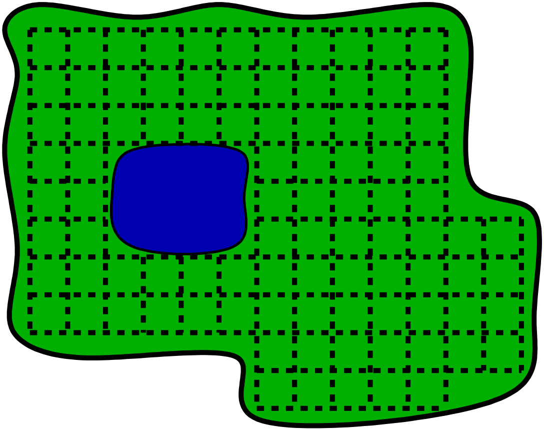
Grid Environments



Environment

- Convenient for motion planning tasks:

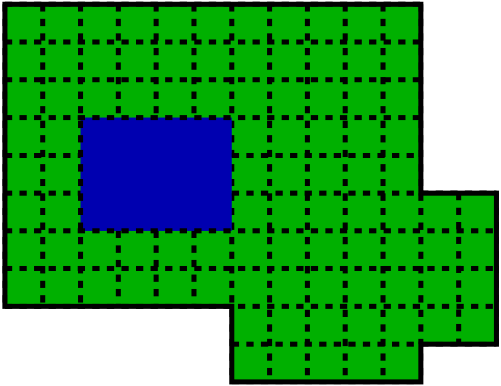
  Subdivide env. by integer grid
- E.g.: cell size $\approx$ size of robot's tool
- Simple $\Leftrightarrow$ No holes



Environment

- Convenient for motion planning tasks:

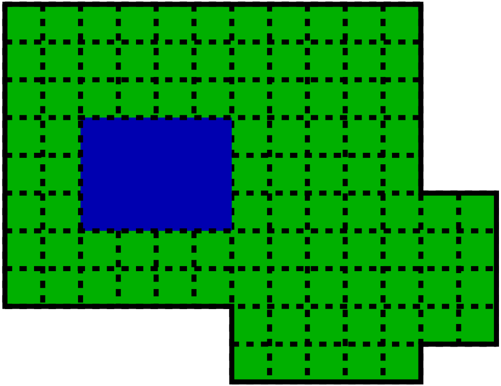
  Subdivide env. by integer grid
- E.g.: cell size ≈ size of robot's tool
- Simple ⇔ No holes



Environment

- Convenient for motion planning tasks:

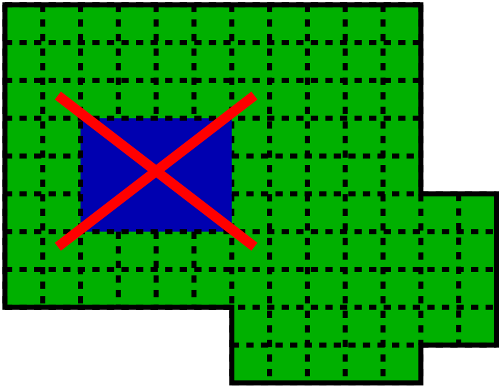
  Subdivide env. by integer grid
- E.g.: cell size ≈ size of robot's tool
- Simple ⇔ No holes



Environment

- Convenient for motion planning tasks:

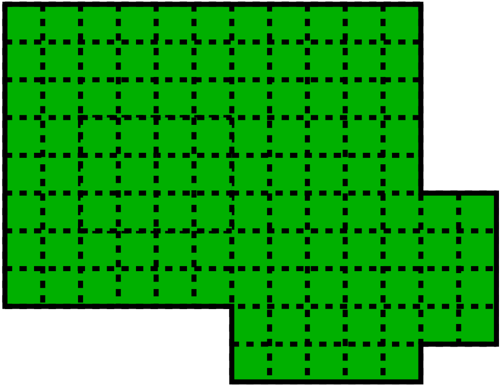
  Subdivide env. by integer grid
- E.g.: cell size ≈ size of robot's tool
- Simple ⇔ No holes



Environment

- Convenient for motion planning tasks:

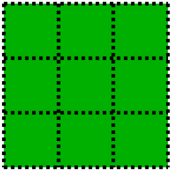
  Subdivide env. by integer grid
- E.g.: cell size $\approx$ size of robot's tool
- Simple $\Leftrightarrow$ No holes



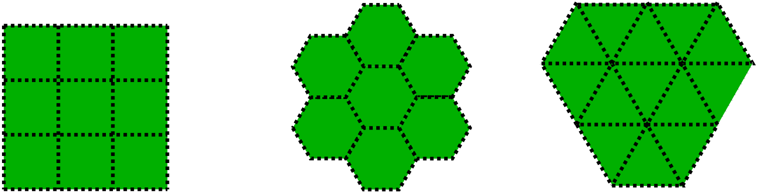
Environment

- Convenient for motion planning tasks:

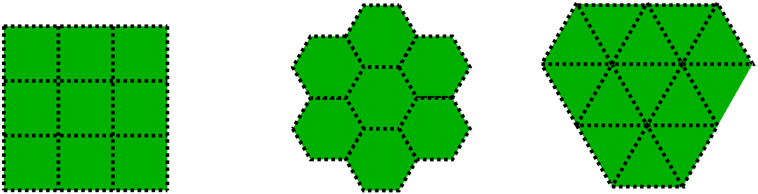
  Subdivide env. by integer grid
- E.g.: cell size $\approx$ size of robot's tool
- Simple $\Leftrightarrow$ No holes



- Usually: Square grids
- Other regular tilings: hexagonal / triangular grids
- Agent:
  - No vision
  - Sense adjacent cells
  - Move to free, adjacent cell

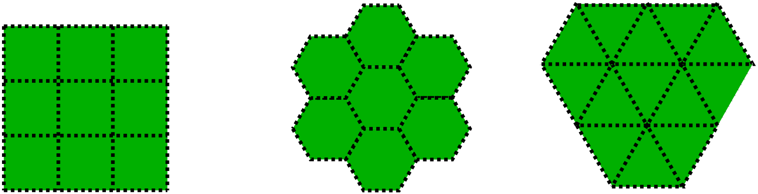


- Usually: Square grids
- Other regular tilings: hexagonal / triangular grids
- Agent:
  - — No vision
  - — Sense adjacent cells
  - — Move to free, adjacent cell



- Usually: Square grids
- Other regular tilings: hexagonal / triangular grids
- Agent:
  - — No vision
  - — Sense adjacent cells
  - — Move to free, adjacent cell

Other Grid Types

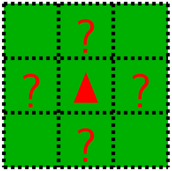


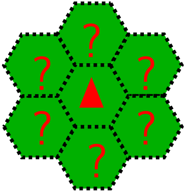
- Usually: Square grids
- Other regular tilings: hexagonal / triangular grids
- Agent:

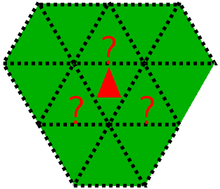
  — No vision

  — Sense adjacent cells

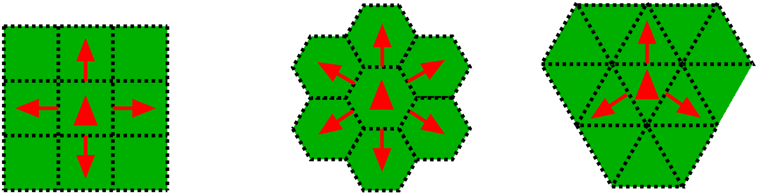
  — Move to free, adjacent cell







- Usually: Square grids
- Other regular tilings: hexagonal / triangular grids
- Agent:
  - — No vision
  - — Sense adjacent cells
  - — Move to free, adjacent cell



- Usually: Square grids
- Other regular tilings: hexagonal / triangular grids
- Agent:
  — No vision
  — Sense adjacent cells
  — Move to free, adjacent cell

The Task

Online exploration (or covering):

- Given: an unknown grid environment, P
  start cell, s, along the boundary
- Task: Find a tour that
  - visits every cell of P at least once
  - returns to the start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

Online exploration (or covering):

- Given: an unknown grid environment, P
  start cell, s, along the boundary
- Task: Find a tour that
  - visits every cell of P at least once
  - returns to the start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

Online exploration (or covering):

- Given: an unknown grid environment, P
  start cell, s, along the boundary
- Task: Find a tour that
  - visits every cell of P at least once
  - returns to the start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

The Task

Online exploration (or covering):

- Given: an unknown grid environment, P
  start cell, s, along the boundary
- Task: Find a tour that
  - visits every cell of P at least once
  - returns to the start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

The Task

Online exploration (or covering):

- Given: an unknown grid environment, P
  start cell, s, along the boundary
- Task: Find a tour that
  - visits every cell of P at least once
  - returns to the start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

Online exploration (or covering):

- Given: an unknown grid environment, P
  start cell, s, along the boundary
- Task: Find a tour that
  - visits every cell of P at least once
  - returns to the start point
  - can be computed online
  - is as short as possible
- For example: lawn mowing, cleaning

Previous Work (square grids)

Offline exploration (environment is known in advance)

- With holes: NP-hard [Itai, Papadimitriou, Szwarcfiter; 1982]
- Approx. [Ntafos; 1992] [Arkin, Fekete, Mitchell; 2000]

Online exploration

- With holes:
  [Icking, Kamphans, Klein, Langetepe; 2000]: 2-competitive
  [Gabriely, Rimon; 2000]
- Without holes:
  [Icking, Kamphans, Klein, Langetepe; 2005]: $\frac{4}{3}$-competitive

Previous Work (square grids)

Offline exploration (environment is known in advance)

- With holes: NP-hard [Itai, Papadimitriou, Szwarcfiter; 1982]
- Approx. [Ntafos; 1992] [Arkin, Fekete, Mitchell; 2000]

Online exploration

- With holes:
  [Icking, Kamphans, Klein, Langetepe; 2000]: 2-competitive
  [Gabriely, Rimon; 2000]
- Without holes:
  [Icking, Kamphans, Klein, Langetepe; 2005]: $\frac{4}{3}$-competitive

Problem:

Online exploration of simple hexagonal/ triangular grid polygons

Theorem

No online exploration strategy achieves a competitive factor better than

- $\frac{7}{6}$ *in simple* ***triangular*** *grid polygon.*
- $\frac{14}{13}$ *in simple* ***hexagonal*** *grid polygon.*

Lower bound for polygons with holes: 2 [IKKL 2000]

Problem:

Online exploration of simple hexagonal/ triangular grid polygons

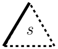
Theorem

No online exploration strategy achieves a competitive factor better than

- $\frac{7}{6}$ *in simple* ***triangular*** *grid polygon.*
- $\frac{14}{13}$ *in simple* ***hexagonal*** *grid polygon.*

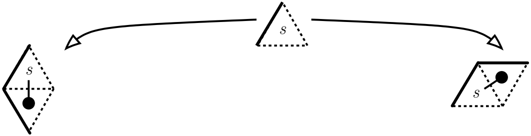
Lower bound for polygons with holes: 2 [IKKL 2000]

Proof: Lower Bound (Triangular Grids)



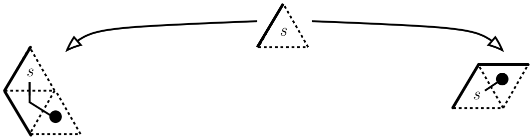
Proof: Lower Bound (Triangular Grids)

South or East



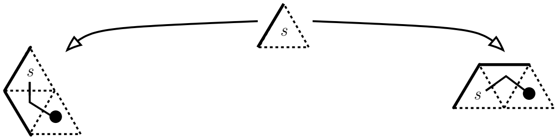
Proof: Lower Bound (Triangular Grids)

East



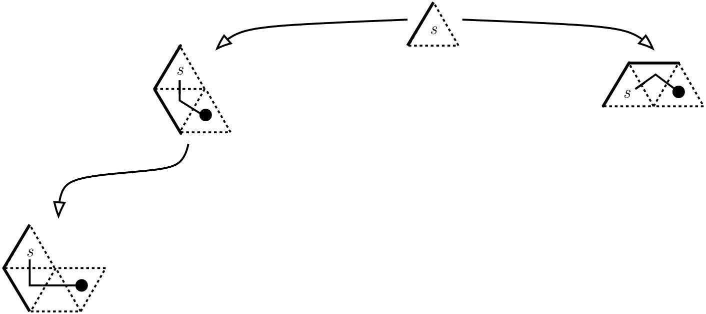
Proof: Lower Bound (Triangular Grids)

East



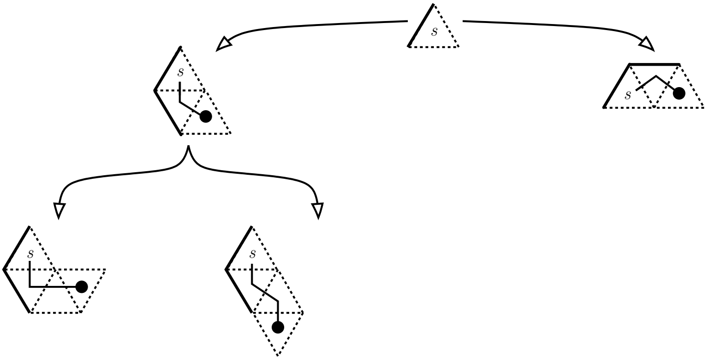
Proof: Lower Bound (Triangular Grids)

2 Possibilities: South



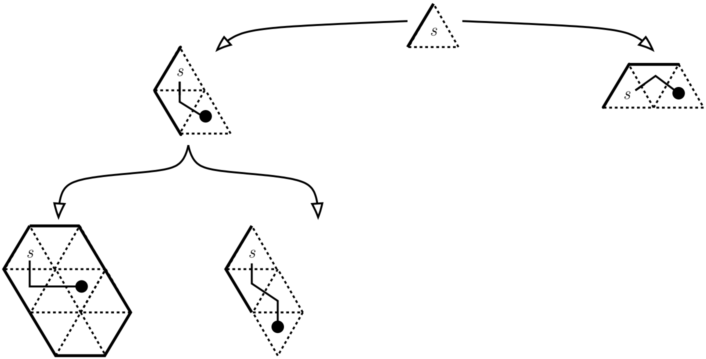
Proof: Lower Bound (Triangular Grids)

2 Possibilities: South, East



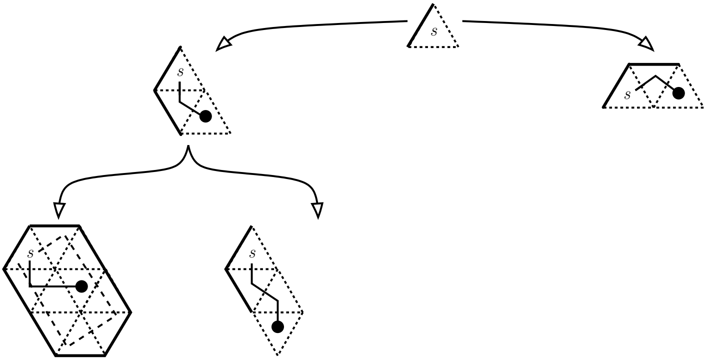
Proof: Lower Bound (Triangular Grids)

Close polygon



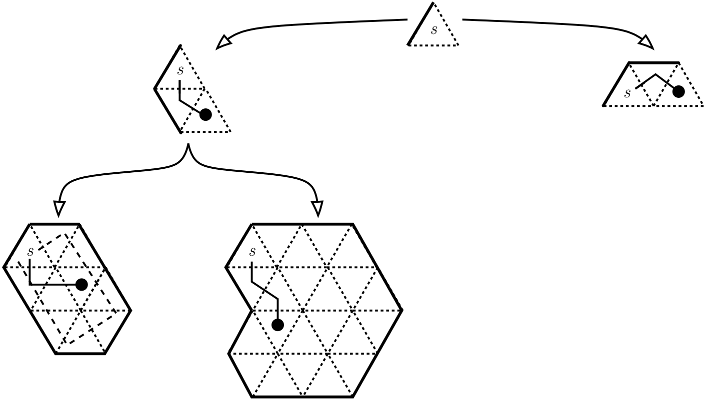
Proof: Lower Bound (Triangular Grids)

Online vs. Optimal



$\frac{12}{10}$

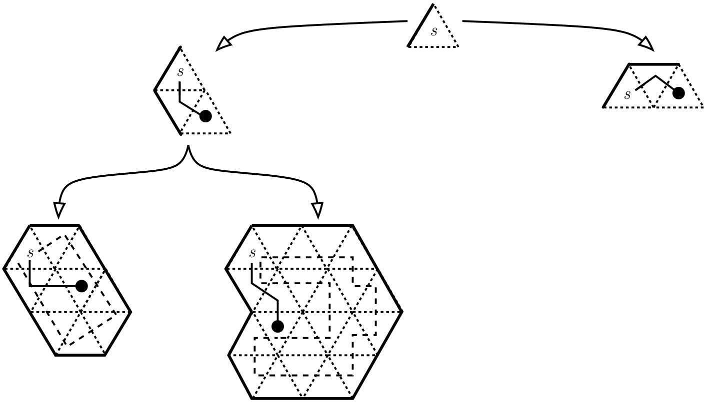
Close polygon



$\frac{12}{10}$

Proof: Lower Bound (Triangular Grids)

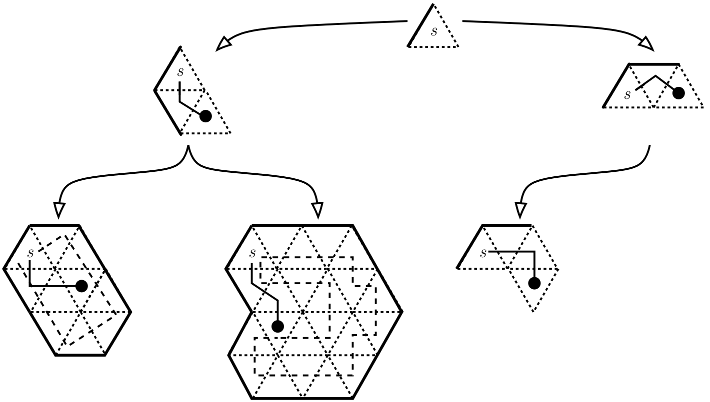
Online vs. Optimal



$\frac{12}{10}$ $\frac{26}{22}$

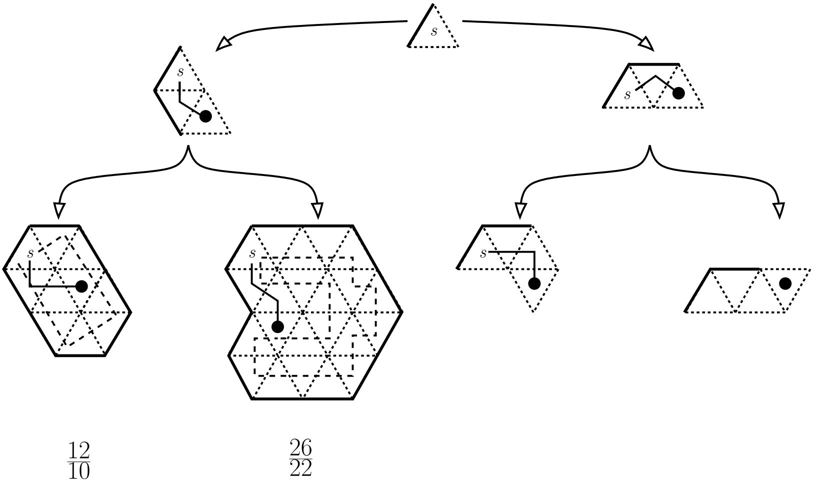
Proof: Lower Bound (Triangular Grids)

2 Possibilities: South



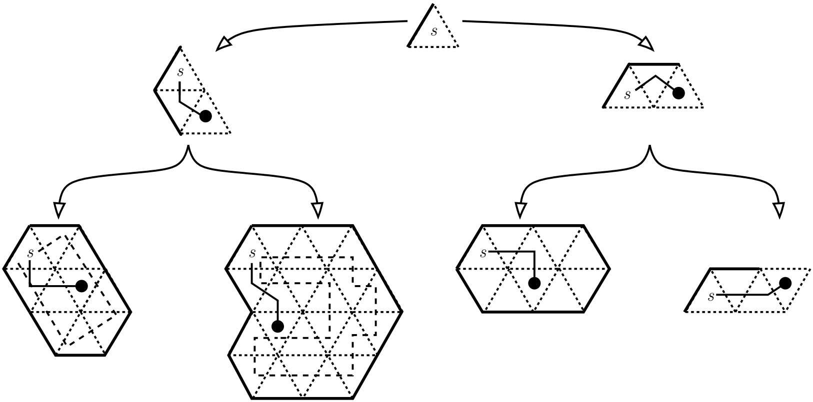
$\frac{12}{10}$ $\frac{26}{22}$

2 Possibilities: South, East



Proof: Lower Bound (Triangular Grids)

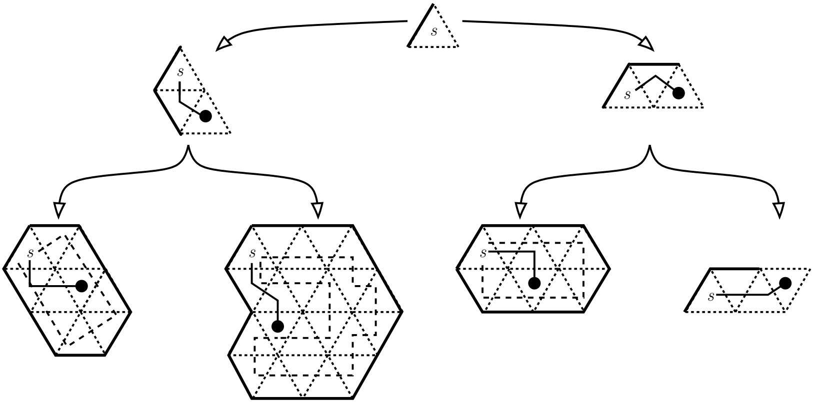
Close polygon



$\frac{12}{10}$ $\frac{26}{22}$

Proof: Lower Bound (Triangular Grids)

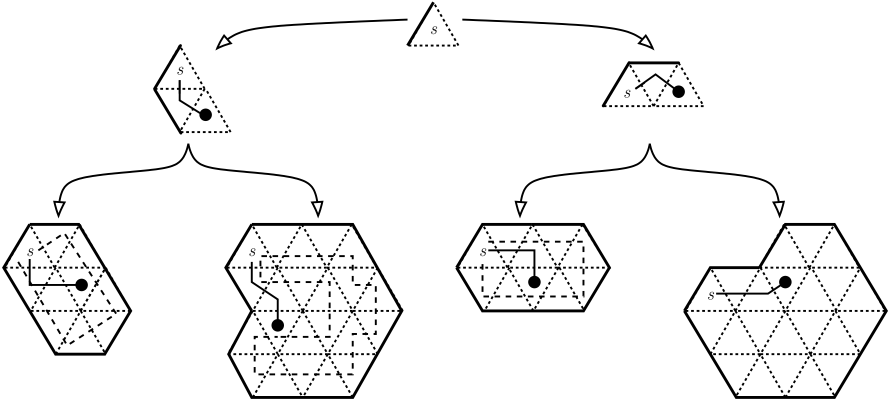
Online vs. Optimal



$\frac{12}{10}$ $\frac{26}{22}$ $\frac{12}{10}$

Proof: Lower Bound (Triangular Grids)

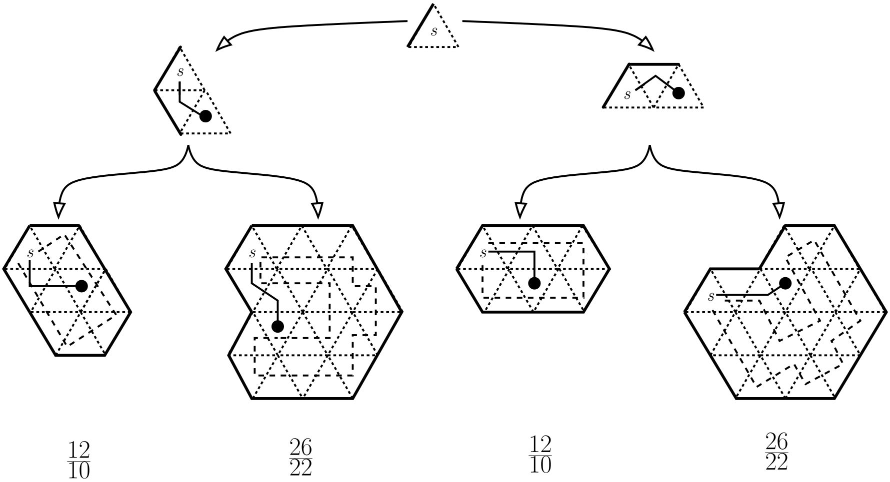
Close polygon



$\frac{12}{10}$ $\frac{26}{22}$ $\frac{12}{10}$

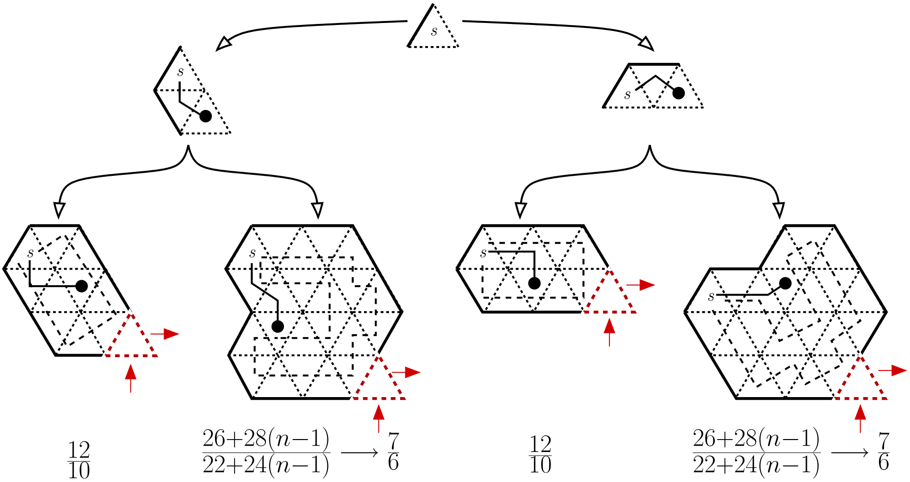
Proof: Lower Bound (Triangular Grids)

Online vs. Optimal

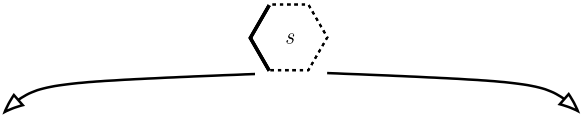


Proof: Lower Bound (Triangular Grids)

Polygons of arbitrary size

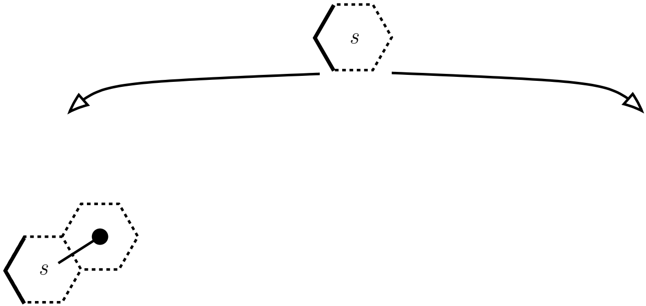


Proof: Lower Bound (Hexagonal Grids)



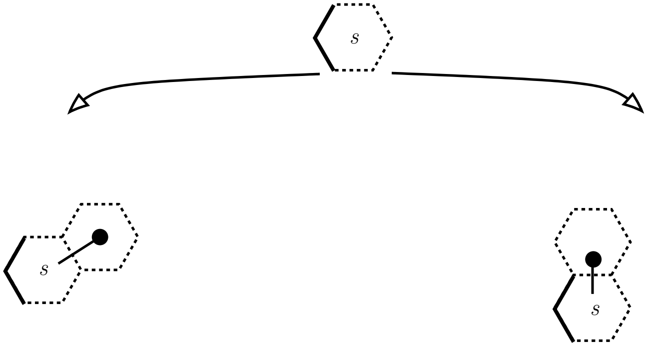
Proof: Lower Bound (Hexagonal Grids)

Leave boundary



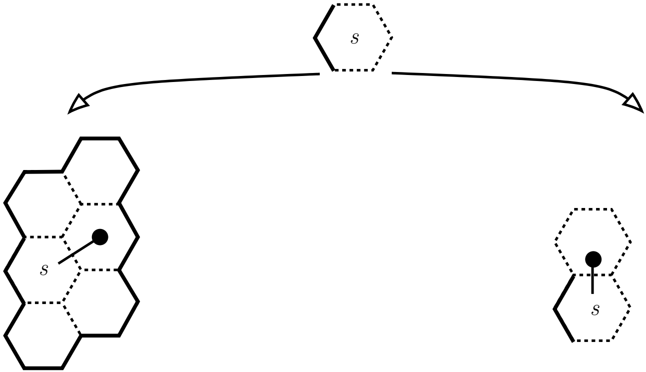
Proof: Lower Bound (Hexagonal Grids)

Follow boundary



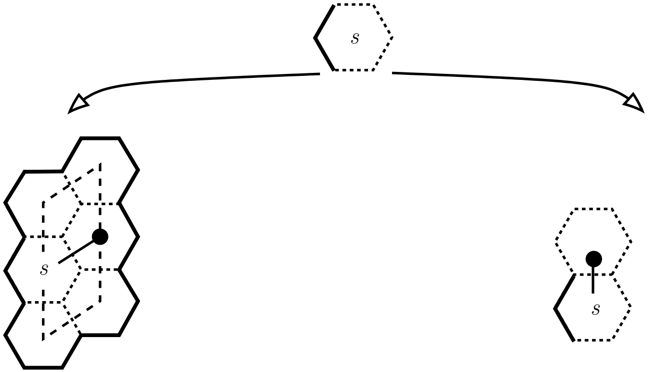
Proof: Lower Bound (Hexagonal Grids)

Close block



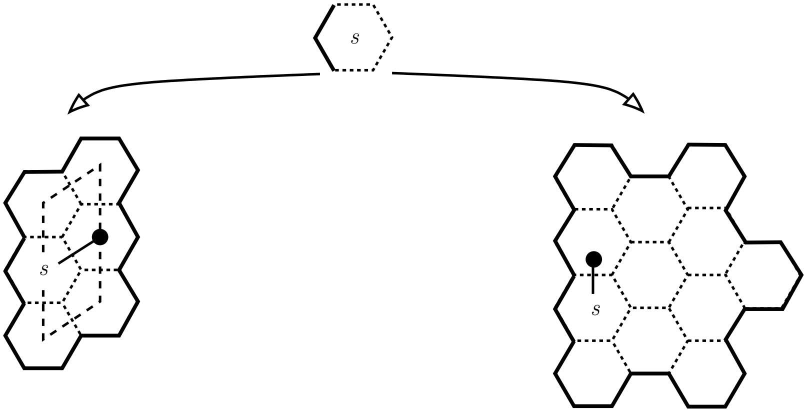
Proof: Lower Bound (Hexagonal Grids)

Online vs. Optimal



$$\frac{7}{6}$$

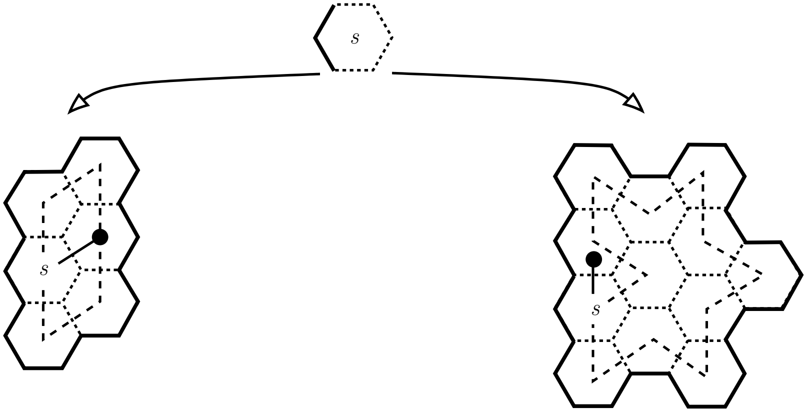
Close block



$\frac{7}{6}$

Proof: Lower Bound (Hexagonal Grids)

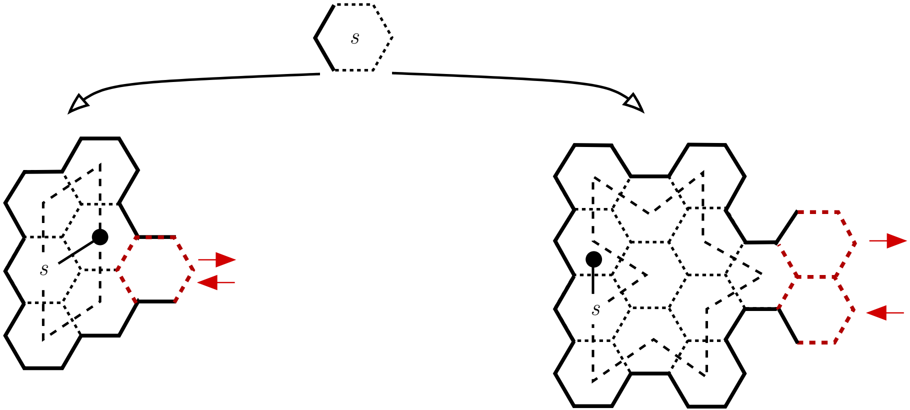
Online vs. Optimal



$\frac{7}{6}$ $\qquad$ $\frac{13}{12}$

Proof: Lower Bound (Hexagonal Grids)

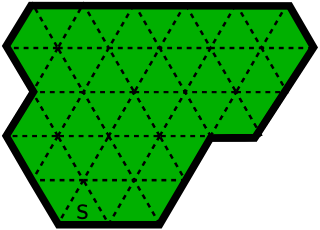
Polygons of arbitrary size

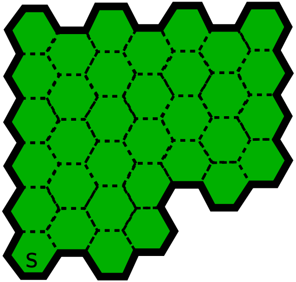


$\frac{7}{6}$

$$\frac{13+14(n-1)}{12+13(n-1)} \longrightarrow \frac{14}{13}$$

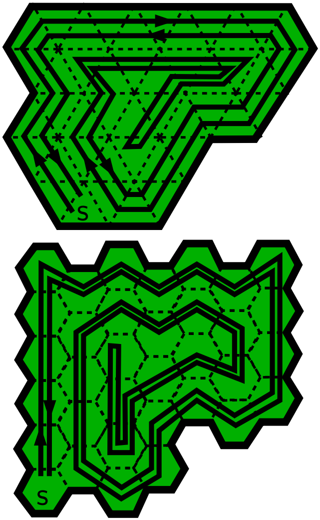
SmartDFS: An exploration strategy (1)





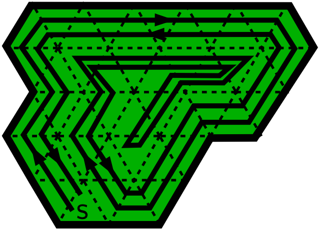
- First idea: Apply depth-first search (DFS)
- *Left-hand rule*: keep boundary and visited cell on the left side.
- Visits *each* cell twice!

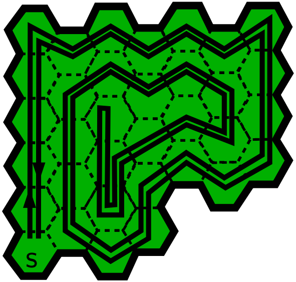
SmartDFS: An exploration strategy (1)



- First idea: Apply depth-first search (DFS)
- *Left-hand rule*: keep boundary and visited cell on the left side.
- Visits *each* cell twice!

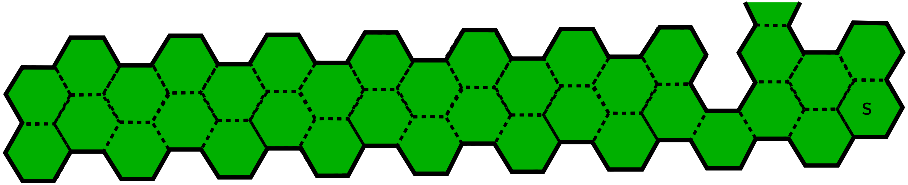
SmartDFS: An exploration strategy (1)





- First idea: Apply depth-first search (DFS)
- *Left-hand rule*: keep boundary and visited cell on the left side.
- Visits *each* cell twice!

SmartDFS: An exploration strategy (2)

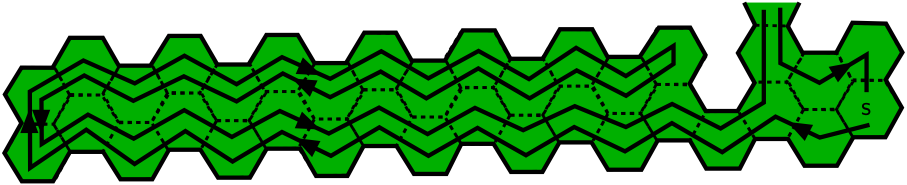


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (2)

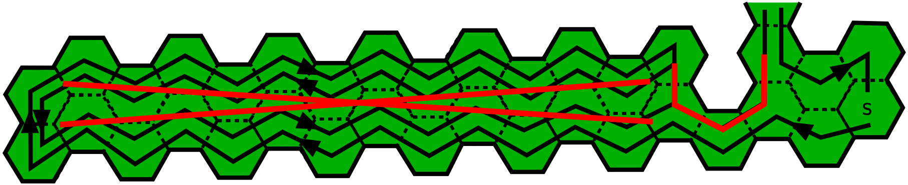


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (2)

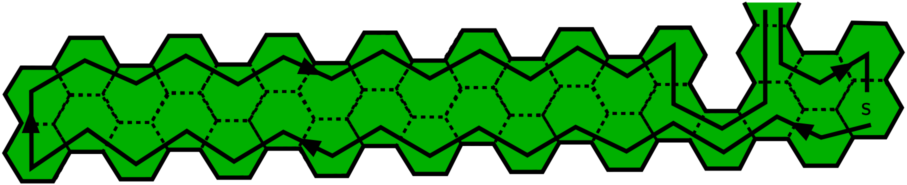


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (2)

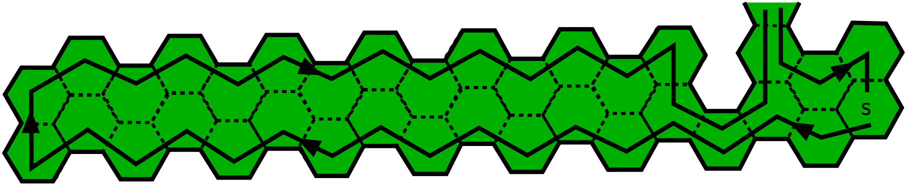


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (2)

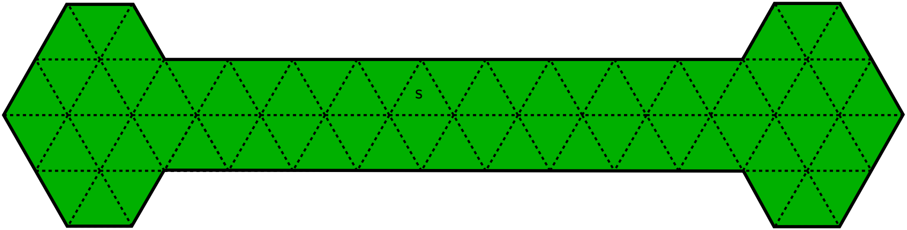


- DFS visits each cell twice
- More reasonable: Return directly to unvisited cell
- Improved DFS

Improvement 1

Return directly to those cells that have unexplored neighbors.

SmartDFS: An exploration strategy (3)

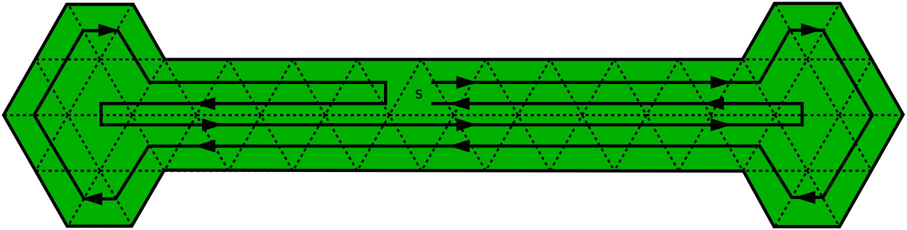


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

SmartDFS: An exploration strategy (3)

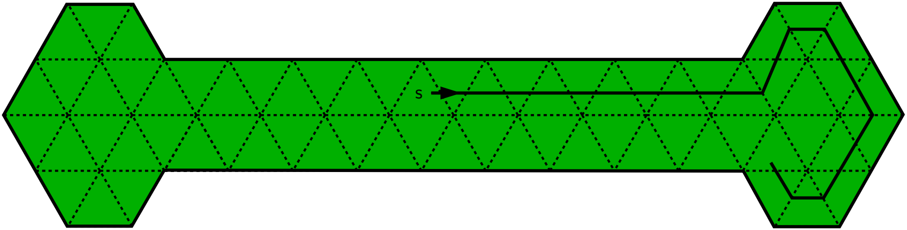


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

SmartDFS: An exploration strategy (3)

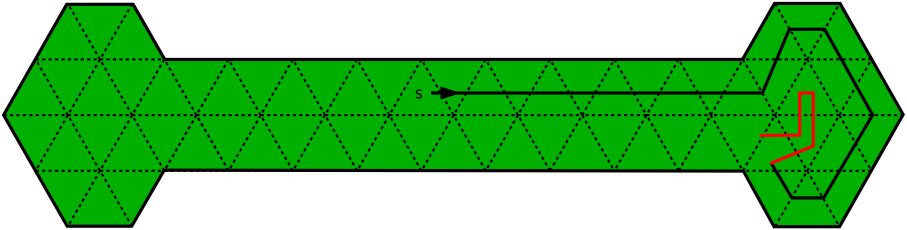


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

SmartDFS: An exploration strategy (3)

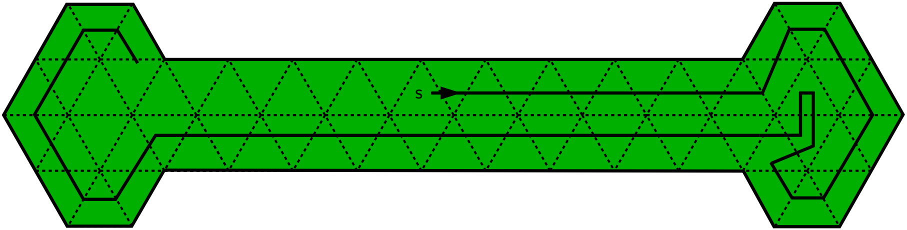


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

SmartDFS: An exploration strategy (3)

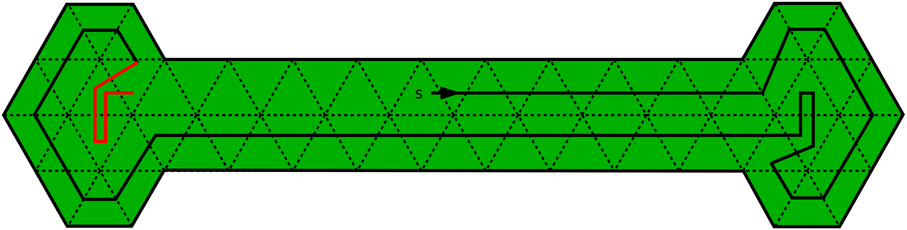


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to s
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

SmartDFS: An exploration strategy (3)

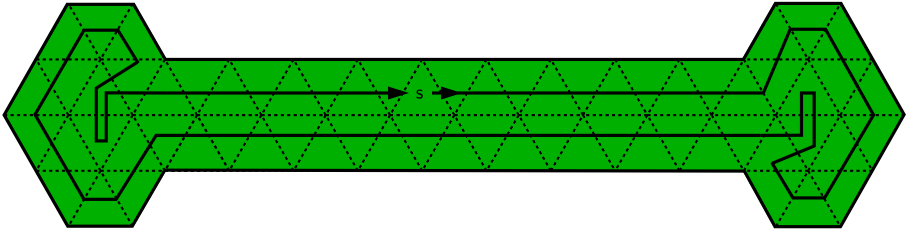


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

SmartDFS: An exploration strategy (3)

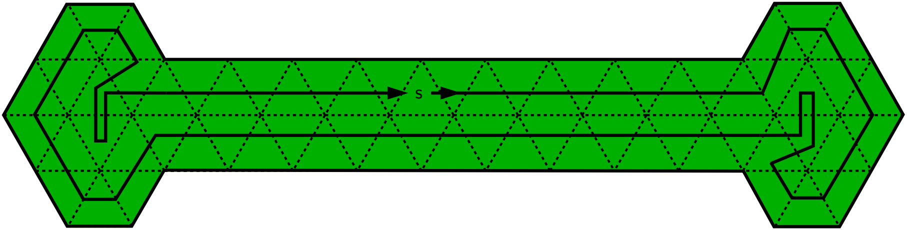


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

SmartDFS: An exploration strategy (3)

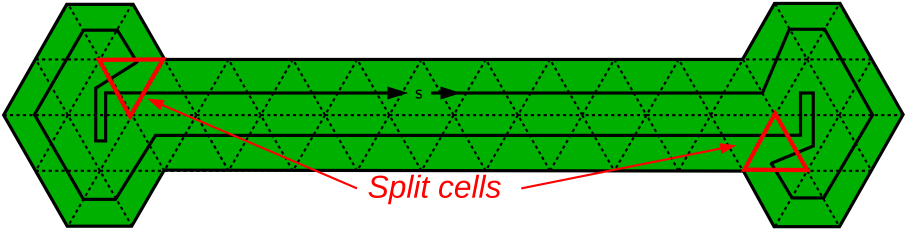


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

SmartDFS: An exploration strategy (3)

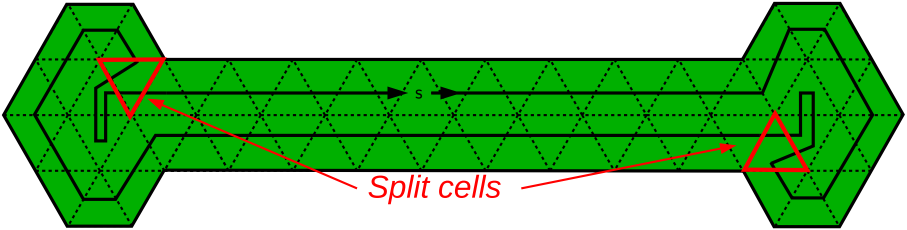


- DFS visits long corridor four times
- More reasonable: Visit right part immediately, continue with the corridor, visit left part, return to *s*
- Long corridor is traversed only two times!
- *Split cells*: Set of unvisited cells gets disconnected

Improvement 2

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

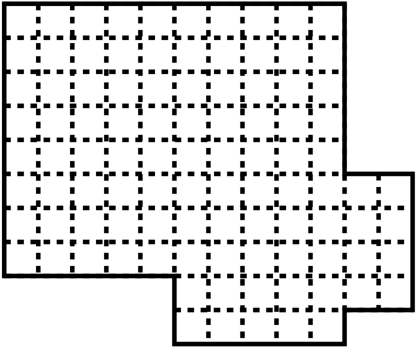
SmartDFS: An exploration strategy (3)



- 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).

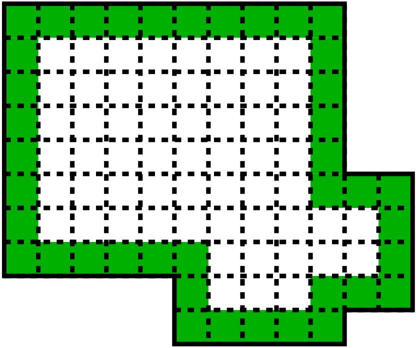


- *First layer* :=
  Boundary cells of P
- *1-offset* :=
  P without first layer
- Analogously: *Second layer*
- *2-offset* and so on
- $E(P)$: #edges between free and blocked cells

Lemma (Number of edges in offsets)

P' is ℓ-offset of $P \Rightarrow E(P') \leq E(P) - 2k\ell$

($k \in \{3, 4, 6\}$ for $\triangle$, $\square$, $\hexagon$).

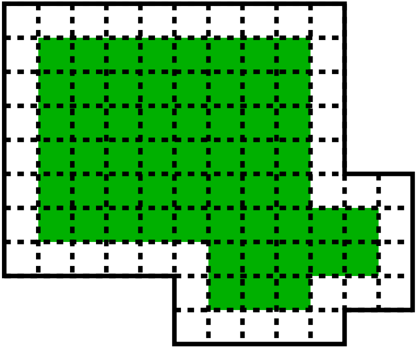


- *First layer* :=
  Boundary cells of P
- *1-offset* :=
  P without first layer
- Analogously: *Second layer*
- *2-offset* and so on
- $E(P)$: #edges between free and blocked cells

Lemma (Number of edges in offsets)

P' is ℓ-offset of $P \Rightarrow E(P') \leq E(P) - 2k\ell$

($k \in \{3, 4, 6\}$ for $\triangle$, $\square$, $\hexagon$).



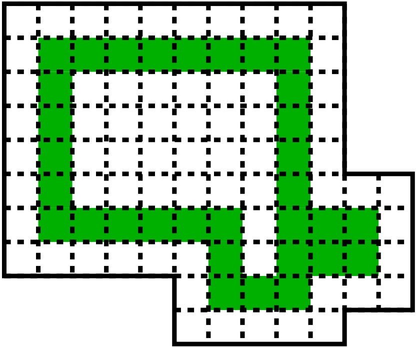
- *First layer* :=
  Boundary cells of P
- *1-offset* :=
  P without first layer
- Analogously: *Second layer*
- *2-offset* and so on
- $E(P)$: #edges between free and blocked cells

Lemma (Number of edges in offsets)

P' is ℓ-offset of $P \Rightarrow E(P') \leq E(P) - 2k\ell$

($k \in \{3, 4, 6\}$ for $\triangle$, $\square$, $\hexagon$).

Layer and Offset

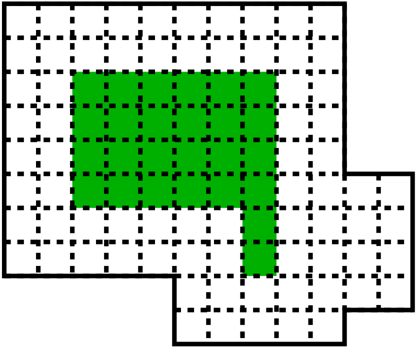


- *First layer* :=
  Boundary cells of P
- *1-offset* :=
  P without first layer
- Analogously: *Second layer*
- *2-offset* and so on
- $E(P)$: #edges between free and blocked cells

Lemma (Number of edges in offsets)

P' is ℓ-offset of $P \Rightarrow E(P') \leq E(P) - 2k\ell$

($k \in \{3, 4, 6\}$ for $\triangle$, $\square$, $\hexagon$).



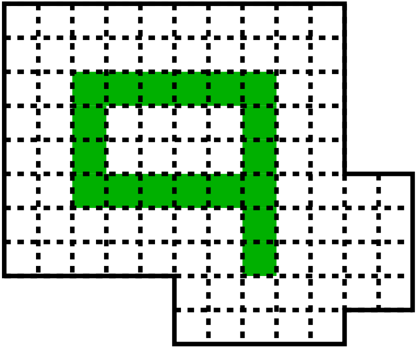
- *First layer* :=
  Boundary cells of P
- *1-offset* :=
  P without first layer
- Analogously: *Second layer*
- *2-offset* and so on
- $E(P)$: #edges between free and blocked cells

Lemma (Number of edges in offsets)

P' is ℓ-offset of $P \Rightarrow E(P') \leq E(P) - 2k\ell$

($k \in \{3, 4, 6\}$ for $\triangle$, $\square$, $\hexagon$).

Layer and Offset

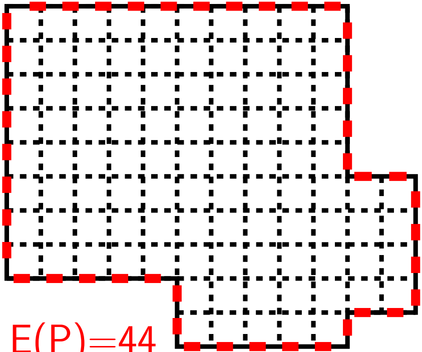


- *First layer* :=
  Boundary cells of P
- *1-offset* :=
  P without first layer
- Analogously: *Second layer*
- *2-offset* and so on
- $E(P)$: #edges between free and blocked cells

Lemma (Number of edges in offsets)

P' is ℓ-offset of $P \Rightarrow E(P') \leq E(P) - 2k\ell$

($k \in \{3, 4, 6\}$ for $\triangle$, $\square$, $\hexagon$).

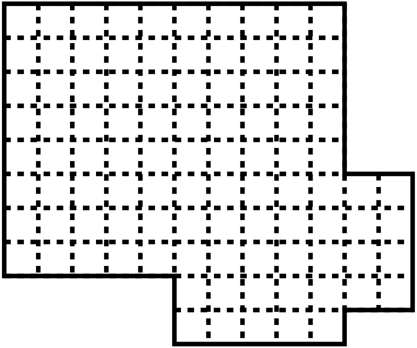


- *First layer* :=
  Boundary cells of P
- *1-offset* :=
  P without first layer
- Analogously: *Second layer*
- *2-offset* and so on
- $E(P)$: #edges between free and blocked cells

Lemma (Number of edges in offsets)

P' is ℓ-offset of $P \Rightarrow E(P') \leq E(P) - 2k\ell$

($k \in \{3, 4, 6\}$ for $\triangle$, $\square$, $\hexagon$).



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

Lemma (Number of edges in offsets)

P' is ℓ-offset of P $\Rightarrow E(P') \leq E(P) - 2k\ell$

($k \in \{3, 4, 6\}$ for $\triangle$, $\square$, $\hexagon$).

Lemma (Shortest Path)

Shortest path between two cells in P:

- $sp(s,t) \leq E(P) - 3$ *(triangular grids)*
- $sp(s,t) \leq \frac{1}{4}E(P) - \frac{3}{2}$ *(hexagonal grids)*

Proof idea.

- Worst case: $\Rightarrow s, t$ in the first layer
  $\Rightarrow$ Path length $\leq \frac{1}{2} \cdot \#$cells in the first layer
- Charge cells in the first layer against $E(P)$

Lemma (Shortest Path)

Shortest path between two cells in P:

- $sp(s,t) \leq E(P) - 3$ *(triangular grids)*
- $sp(s,t) \leq \frac{1}{4}E(P) - \frac{3}{2}$ *(hexagonal grids)*

Proof idea.

- Worst case: – s, t in the first layer
  – Path length $\leq \frac{1}{2} \cdot$ #cells in the first layer
- Charge cells in the first layer against $E(P)$

$\square$

Lemma (Shortest Path)

Shortest path between two cells in P:

- $sp(s,t) \leq E(P) - 3$ *(triangular grids)*
- $sp(s,t) \leq \frac{1}{4}E(P) - \frac{3}{2}$ *(hexagonal grids)*

Proof idea.

- Worst case: – s, t in the first layer
  – Path length $\leq \frac{1}{2} \cdot$ #cells in the first layer
- Charge cells in the first layer against $E(P)$

$\square$

Lemma (Shortest Path)

Shortest path between two cells in P:

- $sp(s,t) \leq E(P) - 3$ *(triangular grids)*
- $sp(s,t) \leq \frac{1}{4}E(P) - \frac{3}{2}$ *(hexagonal grids)*

Proof idea.

- Worst case: – s, t in the first layer
  – Path length $\leq \frac{1}{2} \cdot$ #cells in the first layer
- Charge cells in the first layer against $E(P)$

□

Lemma (Shortest Path)

Shortest path between two cells in P:

- $sp(s,t) \leq E(P) - 3$ *(triangular grids)*
- $sp(s,t) \leq \frac{1}{4}E(P) - \frac{3}{2}$ *(hexagonal grids)*

Proof idea.

- Worst case: – s, t in the first layer
  – Path length $\leq \frac{1}{2} \cdot$ #cells in the first layer
- Charge cells in the first layer against $E(P)$

$\square$

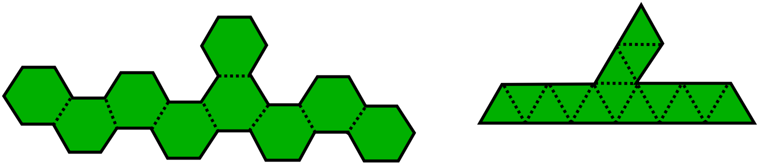
Theorem (Number of Steps)

$$S(P) \leq C(P) + E(P) - 4 \qquad \textit{(Triangular grids)}$$

$$S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2} \qquad \textit{(Hexagonal grids)}$$

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

This bound is exactly achieved in corridors of width 1.



Proof sketch for triangular grids

- $S(P) = \underbrace{C(P)}_{\text{Cells, i.e., necessary steps}} + \underbrace{ex(P)}_{\text{additional cell visits}}$
- Show: $ex(P) \leq E(P) - 4$ (triang.)
- Induction on the number of split cells
- Induction base: No split cell
- Visit every cell in $C(P) - 1$ steps
- Return to s in $\leq E(P) - 3$ steps (Shortest-path lemma)

Proof sketch for triangular grids

- $S(P) = \underbrace{C(P)}_{\text{Cells, i.e., necessary steps}} + \underbrace{ex(P)}_{\text{additional cell visits}}$
- Show: $ex(P) \leq E(P) - 4$ (triang.)
- Induction on the number of split cells
- Induction base: No split cell
-     Visit every cell in $C(P) - 1$ steps
-     Return to s in $\leq E(P) - 3$ steps (Shortest-path lemma)

Proof sketch for triangular grids

- $S(P) = \underbrace{C(P)}_{\text{Cells, i.e., necessary steps}} + \underbrace{ex(P)}_{\text{additional cell visits}}$
- Show: $ex(P) \leq E(P) - 4$ (triang.)
- Induction on the number of split cells
- Induction base: No split cell
- Visit every cell in $C(P) - 1$ steps
- Return to s in $\leq E(P) - 3$ steps (Shortest-path lemma)

Proof sketch for triangular grids

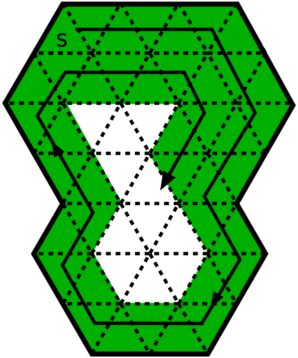
- $S(P) = \underbrace{C(P)}_{\text{Cells, i.e., necessary steps}} + \underbrace{ex(P)}_{\text{additional cell visits}}$
- Show: $ex(P) \leq E(P) - 4$ (triang.)
- Induction on the number of split cells
- Induction base: No split cell
- Visit every cell in $C(P) - 1$ steps
- Return to s in $\leq E(P) - 3$ steps (Shortest-path lemma)

Proof sketch for triangular grids

- $S(P) = \underbrace{C(P)}_{\text{Cells, i.e., necessary steps}} + \underbrace{ex(P)}_{\text{additional cell visits}}$
- Show: $ex(P) \leq E(P) - 4$ (triang.)
- Induction on the number of split cells
- Induction base: No split cell
- Visit every cell in $C(P) - 1$ steps
- Return to s in $\leq E(P) - 3$ steps (Shortest-path lemma)

Proof sketch for triangular grids

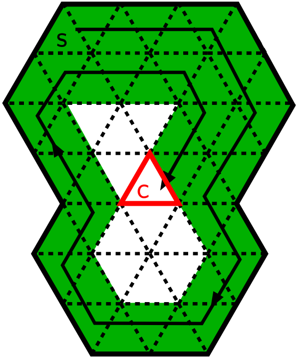
- $S(P) = \underbrace{C(P)}_{\text{Cells, i.e., necessary steps}} + \underbrace{ex(P)}_{\text{additional cell visits}}$
- Show: $ex(P) \leq E(P) - 4$ (triang.)
- Induction on the number of split cells
- Induction base: No split cell
- Visit every cell in $C(P) - 1$ steps
- Return to s in $\leq E(P) - 3$ steps (Shortest-path lemma)



Induction step:

- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \backslash K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$

□

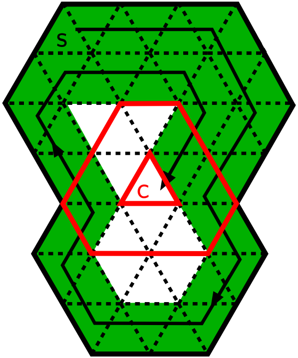


Split cell c

Induction step:

- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$

□



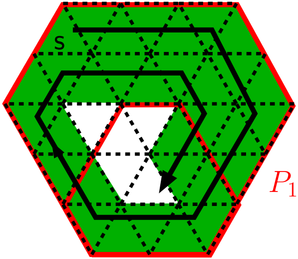
Split cell c

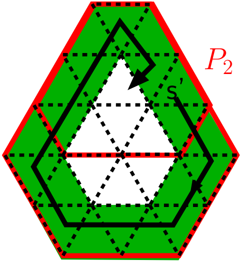
Split polygon Q

Induction step:

- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$

□

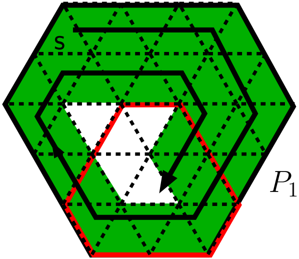


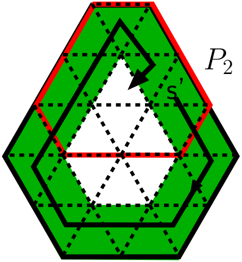


Induction step:

- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$

$\square$

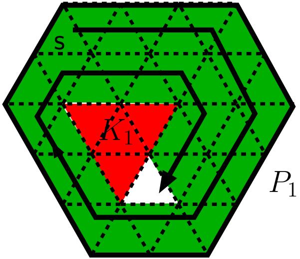


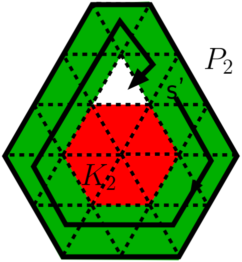


Induction step:

- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$

□

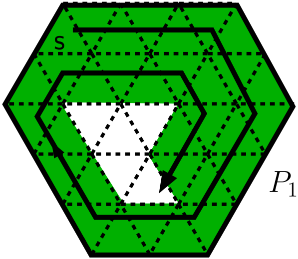


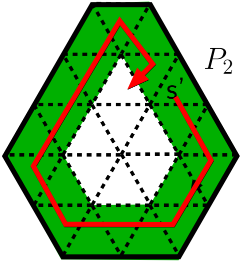


Induction step:

- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$

$\square$

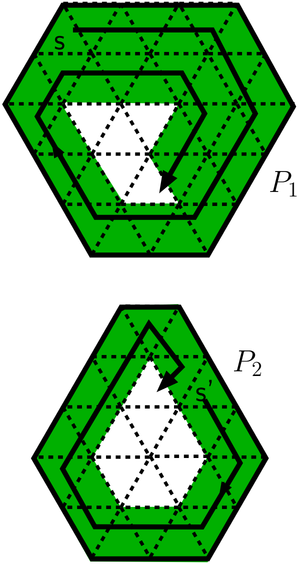




Induction step:

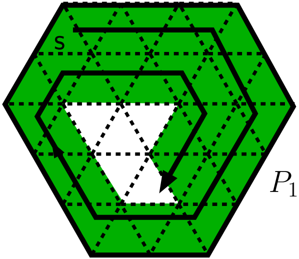
- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$

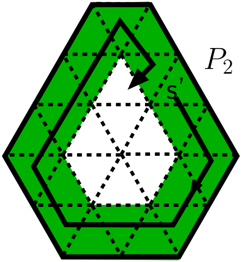
□



Induction step:

- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \backslash K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$





Induction step:

- Explore up to the first split cell, c
- Enlarge c to boundary (add layers)
- Split polygon in two parts: P_1, P_2
- Path outside Q do not change
- K_2 unexplored part of P_2
- Path in $P_2 \setminus K_2$ visits no cell twice
- Apply induction hypothesis to P_1 and $K_2 \cup \{c\}$
- Charge edges of P_1, K_2, Q to $E(P)$

$\square$

Competitivity, hexagonal and triangular grids

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, hexagonal and triangular grids

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, hexagonal and triangular grids

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.

Lemma (Edges in uncritical triangular polygons)

For uncritical grid polygons: $E(P) \leq \frac{1}{3}C(P) + \frac{14}{3}$

Proof idea.

- Successively remove straight line of at least 3 cells (thus at most 1 edge), maintaining the property 'uncritical'
- Ends with 'diamond' polygon, $E(P) = \frac{1}{3}C(P) + \frac{14}{3}$
- Adding removed lines maintains $E(P) \leq \frac{1}{3}C(P) + \frac{14}{3}$ in every step

□

Lemma (Edges in uncritical triangular polygons)

For uncritical grid polygons: $E(P) \leq \frac{1}{3}C(P) + \frac{14}{3}$

Proof idea.

- Successively remove straight line of at least 3 cells (thus at most 1 edge), maintaining the property 'uncritical'
- Ends with 'diamond' polygon, $E(P) = \frac{1}{3}C(P) + \frac{14}{3}$
- Adding removed lines maintains $E(P) \leq \frac{1}{3}C(P) + \frac{14}{3}$ in every step

□

Lemma (Edges in uncritical triangular polygons)

For uncritical grid polygons: $E(P) \leq \frac{1}{3}C(P) + \frac{14}{3}$

Proof idea.

- Successively remove straight line of at least 3 cells (thus at most 1 edge), maintaining the property 'uncritical'
- Ends with 'diamond' polygon, $E(P) = \frac{1}{3}C(P) + \frac{14}{3}$
- Adding removed lines maintains $E(P) \leq \frac{1}{3}C(P) + \frac{14}{3}$ in every step

□

Competitivity in triangular grids

Lemma (Edges in uncritical triangular polygons)

For uncritical grid polygons: $E(P) \leq \frac{1}{3}C(P) + \frac{14}{3}$

Proof idea.

- Successively remove straight line of at least 3 cells (thus at most 1 edge), maintaining the property 'uncritical'
- Ends with 'diamond' polygon, $E(P) = \frac{1}{3}C(P) + \frac{14}{3}$
- Adding removed lines maintains $E(P) \leq \frac{1}{3}C(P) + \frac{14}{3}$ in every step

□

Competitivity (3)

Lemma (Exploration of uncritical polygons)

For uncritical triangular grid polygons: $S(P) \leq C(P) + E(P) - 6$.

Proof sketch

- Shown: $S(P) \leq C(P) + E(P) - 4$
- Used shortest path lemma: $sp(c, s) \leq E(P) - 4$
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Lemma (Exploration of uncritical polygons)

For uncritical triangular grid polygons: $S(P) \leq C(P) + E(P) - 6$.

Proof sketch

- Shown: $S(P) \leq C(P) + E(P) - 4$
- Used shortest path lemma: $sp(c, s) \leq E(P) - 4$
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Lemma (Exploration of uncritical polygons)

For uncritical triangular grid polygons: $S(P) \leq C(P) + E(P) - 6$.

Proof sketch

- Shown: $S(P) \leq C(P) + E(P) - 4$
- Used shortest path lemma: $sp(c, s) \leq E(P) - 4$
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Lemma (Exploration of uncritical polygons)

For uncritical triangular grid polygons: $S(P) \leq C(P) + E(P) - 6$.

Proof sketch

- Shown: $S(P) \leq C(P) + E(P) - 4$
- Used shortest path lemma: $sp(c, s) \leq E(P) - 4$
- 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 triangular grid polygons: $S(P) \leq C(P) + E(P) - 6$.

Proof sketch

- Shown: $S(P) \leq C(P) + E(P) - 4$
- Used shortest path lemma: $sp(c, s) \leq E(P) - 4$
- Proof assumed c, s in the first layer!
- Now: c in the 1-offset
- 2 steps gained!

Lemma (Exploration of uncritical polygons)

For uncritical triangular grid polygons: $S(P) \leq C(P) + E(P) - 6$.

Proof sketch

- Shown: $S(P) \leq C(P) + E(P) - 4$
- Used shortest path lemma: $sp(c, s) \leq E(P) - 4$
- 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.

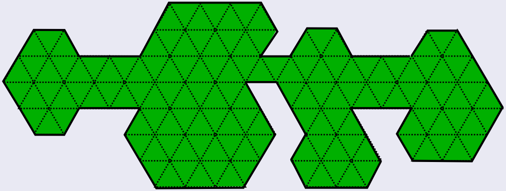
Proof

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

Theorem (Competitivity)

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

Proof

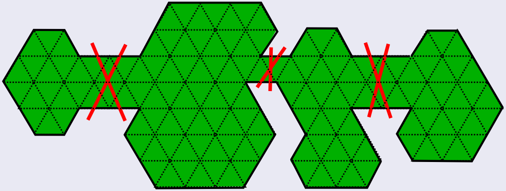


- Remove narrow passages (explored optimally)
- $\Rightarrow$ Split P into P_i
- Consider P_i separately

Theorem (Competitivity)

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

Proof

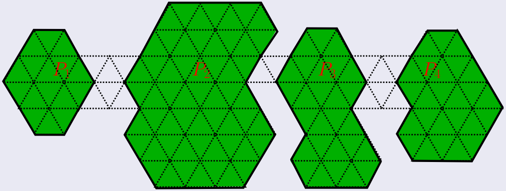


- Remove narrow passages (explored optimally)
- $\Rightarrow$ Split P into P_i
- Consider P_i separately

Theorem (Competitivity)

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

Proof

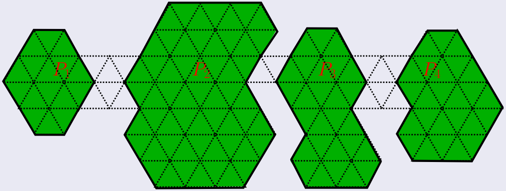


- Remove narrow passages (explored optimally)
- $\Rightarrow$ Split P into P_i
- Consider P_i separately

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

$$\begin{aligned} S(P_i) &\leq C(P_i) + E(P_i) - 6 && \text{by exploration lemma} \\ &\leq C(P_i) + \frac{1}{3}C(P_i) + \frac{14}{3} - 6 && \text{by edges lemma} \\ &= \frac{4}{3}C(P_i) - \frac{4}{3} \end{aligned}$$

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

$$\begin{aligned} S(P_i) &\leq C(P_i) + E(P_i) - 6 && \text{by exploration lemma} \\ &\leq C(P_i) + \frac{1}{3}C(P_i) + \frac{14}{3} - 6 && \text{by edges lemma} \\ &= \frac{4}{3}C(P_i) - \frac{4}{3} \end{aligned}$$

Competitivity Proof: Induction

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

$$\begin{aligned} S(P_i) &\leq C(P_i) + E(P_i) - 6 && \text{by exploration lemma} \\ &\leq C(P_i) + \frac{1}{3}C(P_i) + \frac{14}{3} - 6 && \text{by edges lemma} \\ &= \frac{4}{3}C(P_i) - \frac{4}{3} \end{aligned}$$

Competitivity Proof: Induction

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

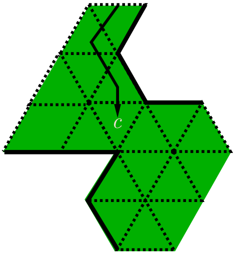
$$\begin{aligned} S(P_i) &\leq C(P_i) + E(P_i) - 6 && \text{by exploration lemma} \\ &\leq C(P_i) + \frac{1}{3}C(P_i) + \frac{14}{3} - 6 && \text{by edges lemma} \\ &= \frac{4}{3}C(P_i) - \frac{4}{3} \end{aligned}$$

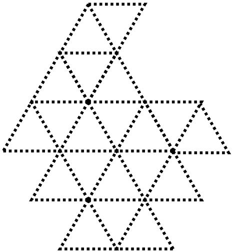
Competitivity Proof: Induction

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

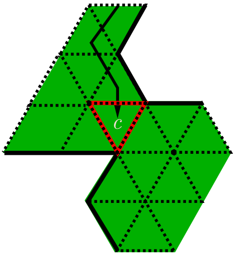
$$
\begin{aligned}
S(P_i) &\leq C(P_i) + E(P_i) - 6 && \text{by exploration lemma} \\
&\leq C(P_i) + \frac{1}{3}C(P_i) + \frac{14}{3} - 6 && \text{by edges lemma} \\
&= \frac{4}{3}C(P_i) - \frac{4}{3}
\end{aligned}
$$

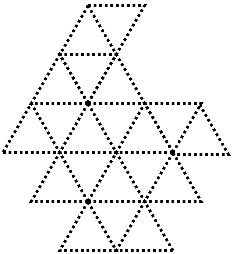
Competitivity Proof: Induction step





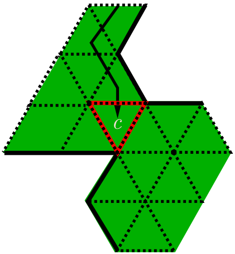
- New component
- $R := \{c\}$
  - Split P_i into P', P''
  - $C(P_i) = C(P') + C(P'') - |R|$
  - $S(P_i) = S(P') + S(P'') - 2(|R| - 1)$
  - Apply induction hypothesis to $S(P')$, $S(P'')$ □
- Explorer meets visited cell c'
- R: shortest path from c to c'

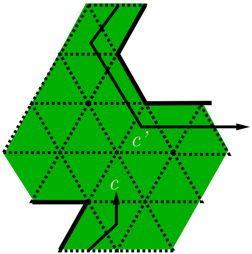




- New component
- $R := \{c\}$
  - Split P_i into P', P''
  - $C(P_i) = C(P') + C(P'') - |R|$
  - $S(P_i) = S(P') + S(P'') - 2(|R| - 1)$
  - Apply induction hypothesis to $S(P')$, $S(P'')$ $\square$
- Explorer meets visited cell c'
- R: shortest path from c to c'

Competitivity Proof: Induction step

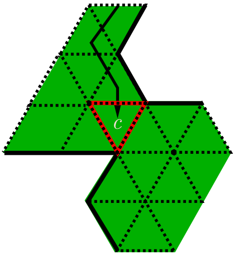


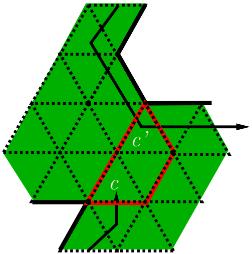


- New component
- $R := \{c\}$
  - Split P_i into P', P''
  - $C(P_i) = C(P') + C(P'') - |R|$
  - $S(P_i) = S(P') + S(P'') - 2(|R| - 1)$
  - Apply induction hypothesis to $S(P')$, $S(P'')$ $\square$

- Explorer meets visited cell c'
- R: shortest path from c to c'

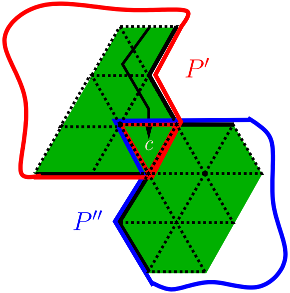
Competitivity Proof: Induction step

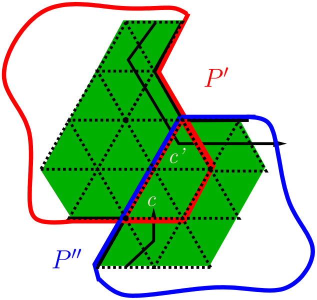




- New component
- $R := \{c\}$
  - Split P_i into P', P''
  - $C(P_i) = C(P') + C(P'') - |R|$
  - $S(P_i) = S(P') + S(P'') - 2\,(|R| - 1)$
  - Apply induction hypothesis to $S(P')$, $S(P'')$ $\square$

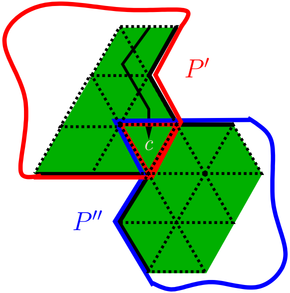
- Explorer meets visited cell c'
- R: shortest path from c to c'

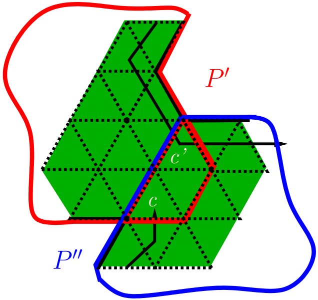




- New component
- $R := \{c\}$
  - Split P_i into P', P''
  - $C(P_i) = C(P') + C(P'') - |R|$
  - $S(P_i) = S(P') + S(P'') - 2(|R| - 1)$
  - Apply induction hypothesis to $S(P')$, $S(P'')$ □

- Explorer meets visited cell c'
- R: shortest path from c to c'

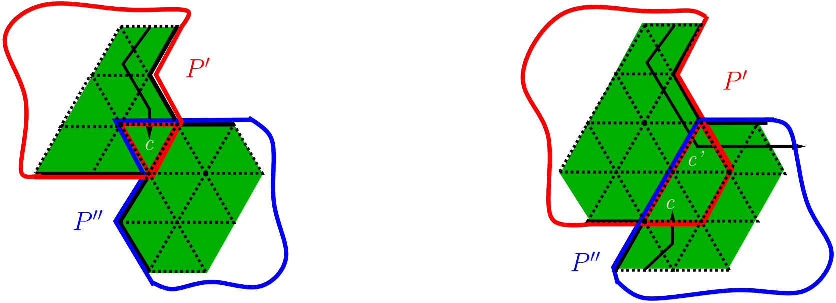




- New component
- $R := \{c\}$
  - Split P_i into P', P''
  - $C(P_i) = C(P') + C(P'') - |R|$
  - $S(P_i) = S(P') + S(P'') - 2(|R| - 1)$
  - Apply induction hypothesis to $S(P')$, $S(P'')$ $\square$

- Explorer meets visited cell c'
- R: shortest path from c to c'

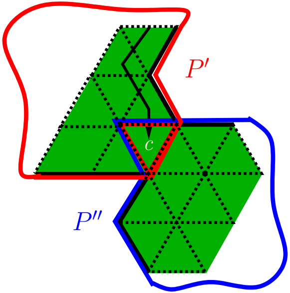
Competitivity Proof: Induction step

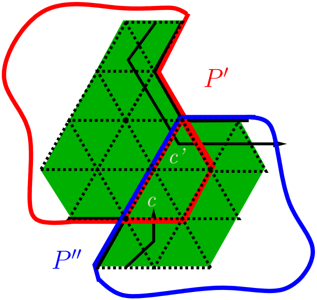


- New component
- $R := \{c\}$
  - Split P_i into P', P''
  - $C(P_i) = C(P') + C(P'') - |R|$
  - $S(P_i) = S(P') + S(P'') - 2\,(|R| - 1)$
  - Apply induction hypothesis to $S(P')$, $S(P'')$ $\square$

- Explorer meets visited cell c'
- R: shortest path from c to c'

Competitivity Proof: Induction step





- New component
- $R := \{c\}$
  - Split P_i into P', P''
  - $C(P_i) = C(P') + C(P'') - |R|$
  - $S(P_i) = S(P') + S(P'') - 2\,(|R| - 1)$
  - Apply induction hypothesis to $S(P')$, $S(P'')$ $\square$
- Explorer meets visited cell c'
- R: shortest path from c to c'

Problem: Online exploration of simple grid polygons

- Lower bound △: $\frac{7}{6}$
- Lower bound ⬡: $\frac{14}{13}$
- Exploration strategy *SmartDFS*
- Upper bound △: $S(P) \leq C(P) + E(P) - 4$
- Upper bound ⬡: $S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2}$
- $\frac{4}{3}$-competitive

- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower bound △: $\frac{7}{6}$
- Lower bound ⬡: $\frac{14}{13}$
- Exploration strategy *SmartDFS*
- Upper bound △: $S(P) \leq C(P) + E(P) - 4$
- Upper bound ⬡: $S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2}$
- $\frac{4}{3}$-competitive

- ToDo: Close the gap!

Summary

Problem: Online exploration of simple grid polygons

- Lower bound △: $\frac{7}{6}$
- Lower bound ⬡: $\frac{14}{13}$
- Exploration strategy *SmartDFS*
- Upper bound △: $S(P) \leq C(P) + E(P) - 4$
- Upper bound ⬡: $S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2}$
- $\frac{4}{3}$-competitive

- ToDo: Close the gap!

Summary

Problem: Online exploration of simple grid polygons

- Lower bound △: $\frac{7}{6}$
- Lower bound ⬡: $\frac{14}{13}$
- Exploration strategy *SmartDFS*
- Upper bound △: $S(P) \leq C(P) + E(P) - 4$
- Upper bound ⬡: $S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2}$
- $\frac{4}{3}$-competitive

- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower bound △: $\frac{7}{6}$
- Lower bound ⬡: $\frac{14}{13}$
- Exploration strategy *SmartDFS*
- Upper bound △: $S(P) \leq C(P) + E(P) - 4$
- Upper bound ⬡: $S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2}$
- $\frac{4}{3}$-competitive

- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower bound △: $\frac{7}{6}$
- Lower bound ⬡: $\frac{14}{13}$
- Exploration strategy *SmartDFS*
- Upper bound △: $S(P) \leq C(P) + E(P) - 4$
- Upper bound ⬡: $S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2}$
- $\frac{4}{3}$-competitive
- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower bound △: $\frac{7}{6}$
- Lower bound ⬡: $\frac{14}{13}$
- Exploration strategy *SmartDFS*
- Upper bound △: $S(P) \leq C(P) + E(P) - 4$
- Upper bound ⬡: $S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2}$
- $\frac{4}{3}$-competitive

- ToDo: Close the gap!

Problem: Online exploration of simple grid polygons

- Lower bound $\triangle$: $\frac{7}{6}$
- Lower bound $\hexagon$: $\frac{14}{13}$
- Exploration strategy *SmartDFS*
- Upper bound $\triangle$: $S(P) \leq C(P) + E(P) - 4$
- Upper bound $\hexagon$: $S(P) \leq C(P) + \frac{1}{4}E(P) - \frac{5}{2}$
- $\frac{4}{3}$-competitive

- ToDo: Close the gap!

Thank you!