Online Motion Planning MA-INF 1314
Smart DFS

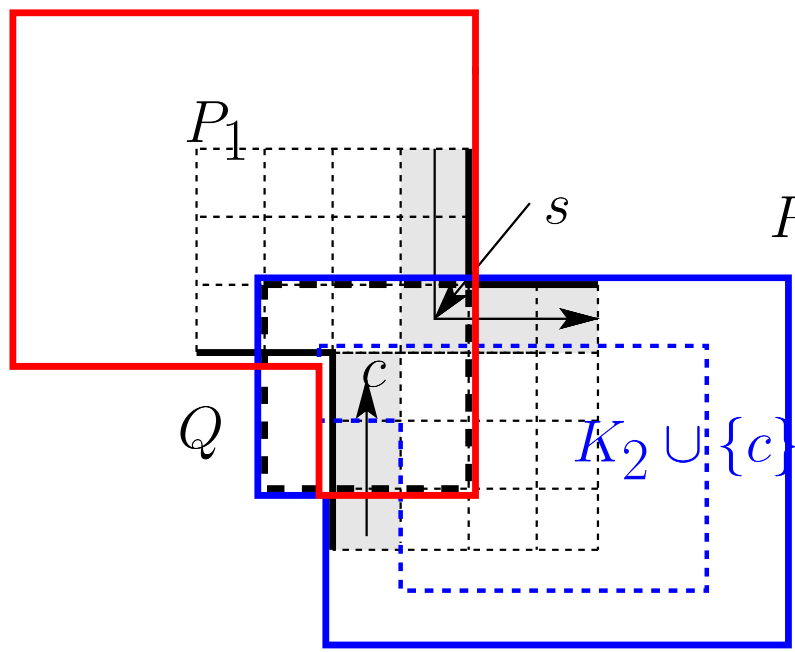
Elmar Langetepe
University of Bonn

Repetition!

- SmartDFS: DFS returnpath and components
- Simple design, complicated analysis: $C + \frac{1}{2}E - 3$
- Structural property: l-Offset, l-Layer
- Edgelemma: l-Offset 8l edges less
- Pathlemma: Shortest path $\leq \frac{1}{2}E(P) - 2$
- Induction: Decompose at splitcell!
- Excesslemma: $\mathit{excess}(P) \leq \mathit{excess}(P_1) + \mathit{excess}(K_2 \cup \{c\}) + 1.$
- Induction over number of splitcells

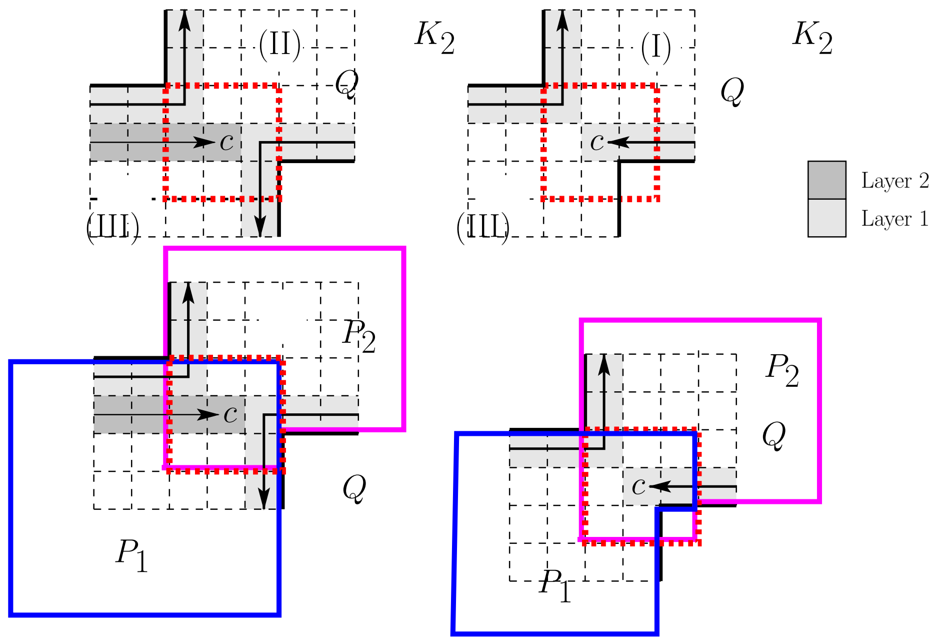
Definition P_1, P_2 and Excesslemma!

- Splitcell c! $K_2 \cup \{c\}$ first!
- P_2 q-Offset of $K_2 \cup \{c\}$ $q = l = 1$, then $P_1 := ((P \setminus P_2) \cup Q) \cap P$!
- *excess*$(P) \leq$ *excess*(P_1) + *excess*$(K_2 \cup \{c\}) + 1$



Repetition: Edges of P und Q

Lemma: P, P_1, P_2 und Q as given. For the number of edges we have $E(P_1) + E(P_2) = E(P) + E(Q)$.



Repetition: Exploration Theorem

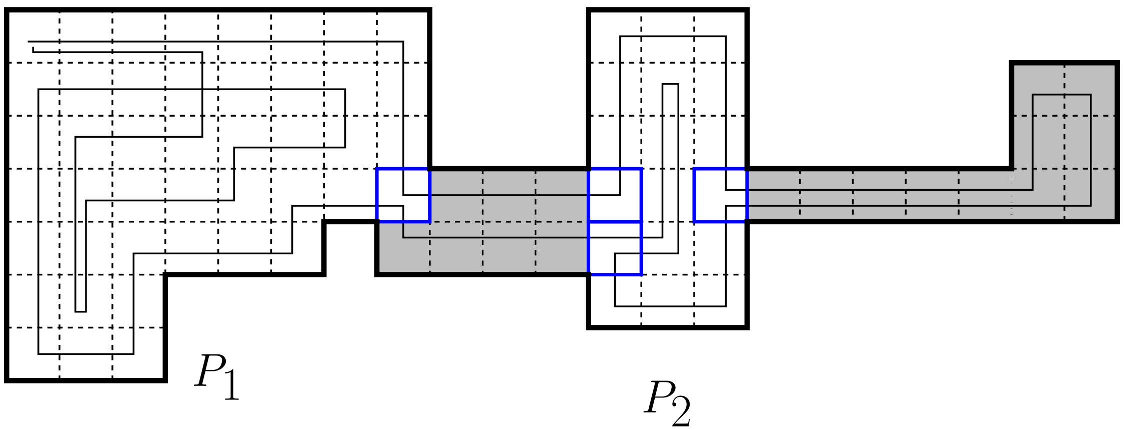
Theorem: SmartDFS explores a simple gridpolygon P with C cells and E boundary edges with at most $C + \frac{1}{2}E - 3$ steps.

Proof: Induction over number of components

- **Induction base**: One component
- Visit cells: $C - 1$, back to start
- Shortest path Lem: $\frac{1}{2}E(P) - 2 + C - 1 = C + \frac{1}{2}E - 3$
- And so on by induction!

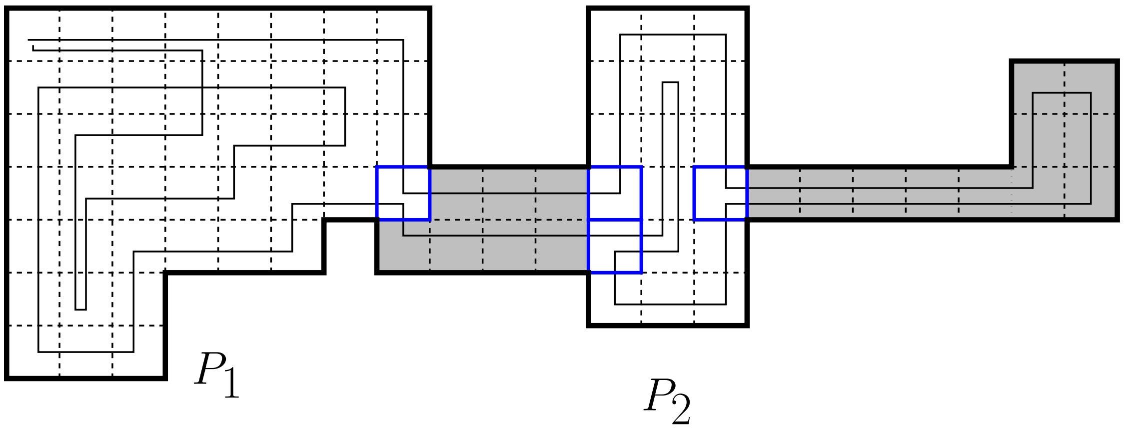
Repetition! Wavefront and competitive ratio

- Wavefront Algorithmu (Lee): $O(n)$, n cells
- Comp. Factor: $S(P) \leq \frac{4}{3}C(P) - 2$ (Lower bound $\frac{7}{6}$)
- Observation: Optimally in narrow passages!



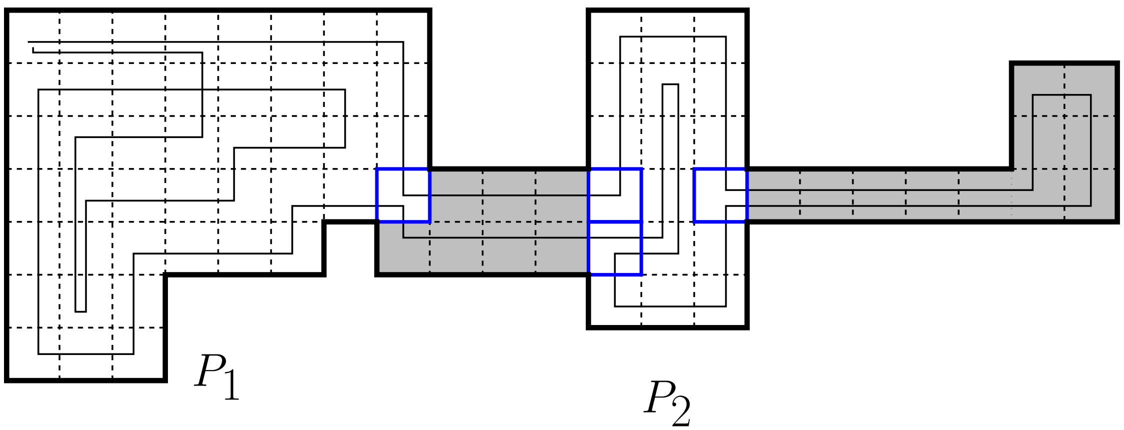
Repetition

- Analyse polygons P_i, $i = 1, \ldots, k$
- Induction over split-cells
- Induction-Base: No split-cell in layer 1.
- **Lemma** $E(P) \leq \frac{2}{3}C(P) + 6$ backward analysis
- **Lemma** $S(P) \leq C(P) + \frac{1}{2}E(P) - 5$, two steps less by Offsetlemma!
- Kombination gives Induction-Base!



Theorem: SmartDFS is $\frac{4}{3}$ competitive

- Narrow passages optimal, sequence of P_i independently!
- Only cells and steps, no edges!!
- Induction in P_i over split-cell number! $S(P_i) \leq \frac{4}{3}C(P_i) - 2$
- Induction base: Use special lemmata!



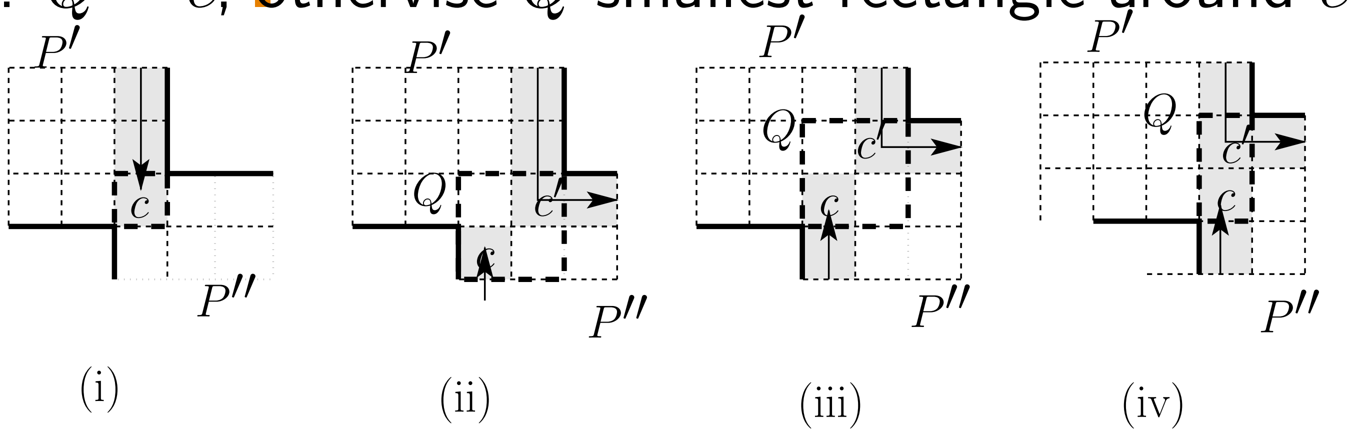
Induction base: $S(P_i) \le \frac{4}{3}C(P_i) - 2$

- P_i no split-cell means, no split-cell in Layer 1
- Apply case-sensitive Lemma: $C(P) + \frac{1}{2}E(P) - 5$
- Apply structural Lemma: $E(P) \le \frac{2}{3}C(P) + 6$

$$
\begin{aligned}
S(P_i) &\le C(P_i) + \frac{1}{2}E(P_i) - 5 \\
&\le C(P_i) + \frac{1}{2}\left(\frac{2}{3}C(P_i) + 6\right) - 5 \\
&= \frac{4}{3}C(P_i) - 2
\end{aligned}
$$

Induction step: $S(P_i) \leq \frac{4}{3}C(P_i) - 2$

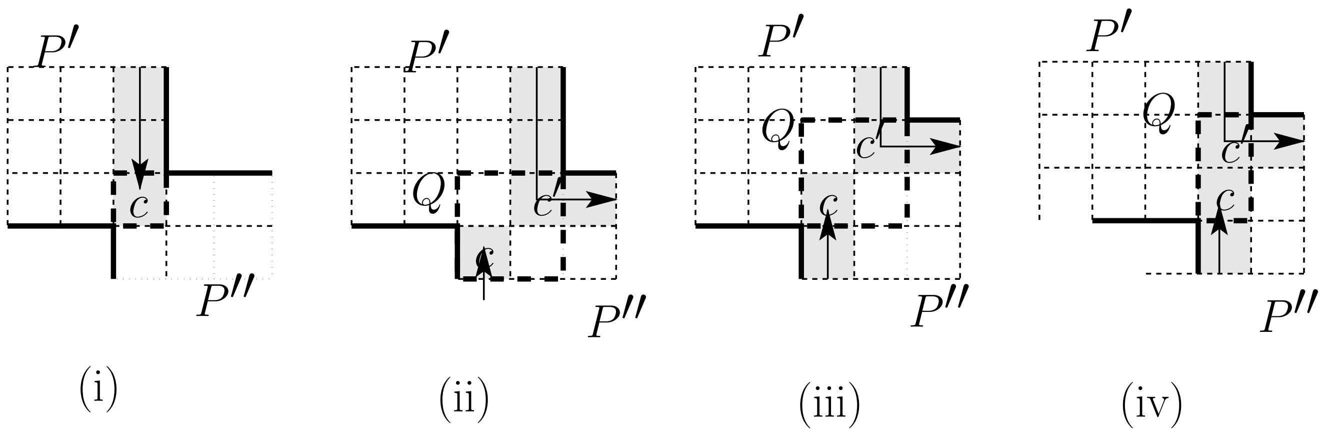
- Split-cell in first layer of P_i, otherwise done: Two Cases
- Split by c adjacent to some c'
- Typ (I) (curr. layer not) or Typ (II) (curr. layer fully.) component
- Split into P' and P'' with Rectangle/Square Q
- Case (i): $Q = c$, othervise Q smallest rectangle around c,c'



Case (i): $S(P_i) \le \frac{4}{3}C(P_i) - 2$

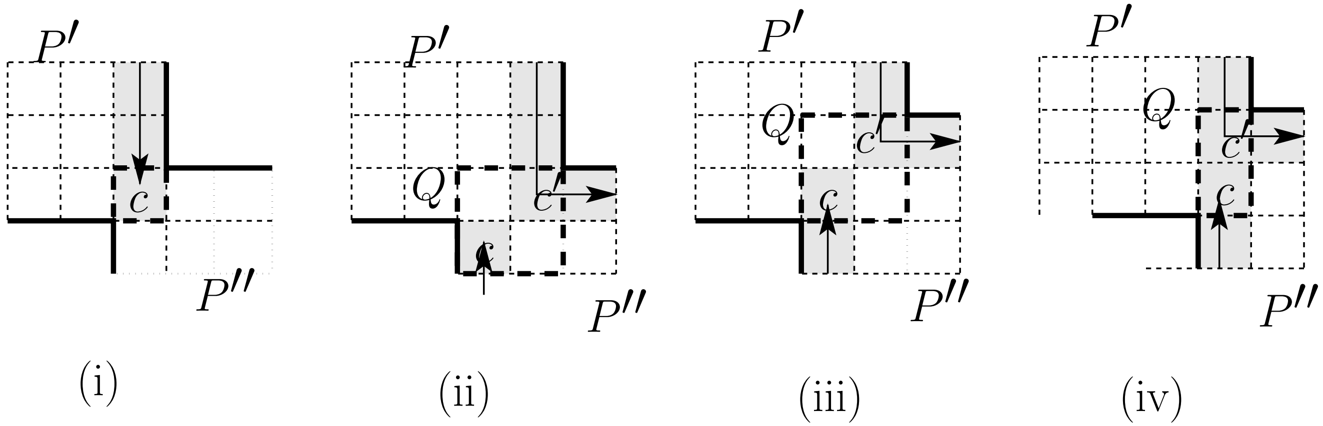
- $S(P_i) = S(P') + S(P'')$ (Gate) $C(P_i) = C(P') + C(P'') - 1$
- Induction: For P' and P'' (less split-cells)

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



Case (ii),(iii): $S(P_i) \leq \frac{4}{3}C(P_i) - 2$

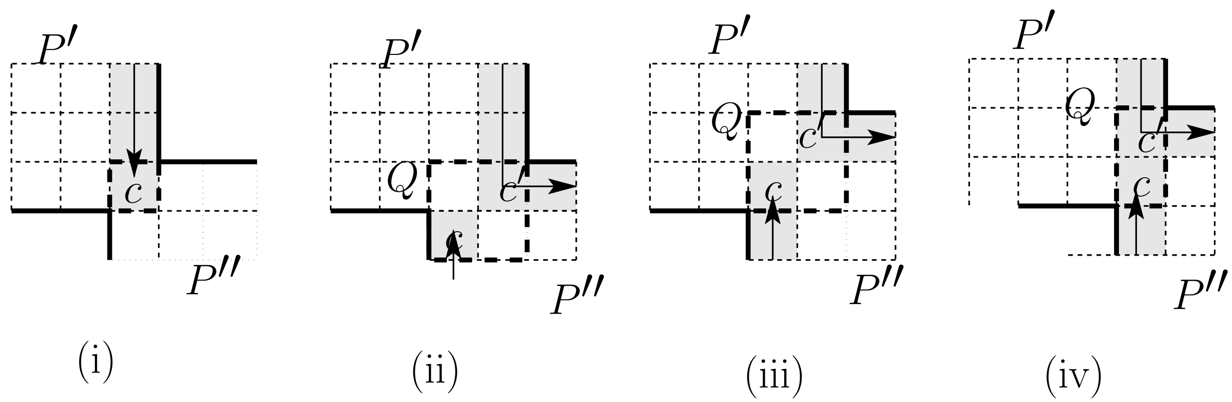
- $|Q| = 4$ but save 4 steps!
- P', P'' separately (I.H.) but
- Path in P_i from c' to c or from c to c' done in P',P''
- Save at least 4$=|Q|$ steps, two cells considered twice!



Case (ii),(iii): $S(P_i) \leq \frac{4}{3}C(P_i) - 2$

- At least $4 = |Q|$ steps less, two cells argument
- $S(P_i) = S(P') + S(P'') - 4$ and $C(P_i) = C(P') + C(P'') - 4$
- Apply I.H. for P' and P''

$$\begin{aligned} S(P_i) &= S(P') + S(P'') - 4 \leq \frac{4}{3}C(P') + \frac{4}{3}C(P'') - 8 \\ &\leq \frac{4}{3}(C(P') + C(P'') - 4) - \frac{8}{3} < \frac{4}{3}C(P_i) - 2 \end{aligned}$$



Summary SmartDFS

- Gridpolygons without holes
- Lower bound: $\frac{7}{6}$
- SmartDFS: $\frac{4}{3}$
- More sophisticated approach: approx. $\frac{5}{4}$
- Lower bound: $\frac{20}{17}$
- Optimal Offline Solution?