Subgoal Graphs for Fast Optimal Pathfinding

Andreas Thüring

Seminar: Search & Optimization
Universität Basel
October 22, 2015

Outline

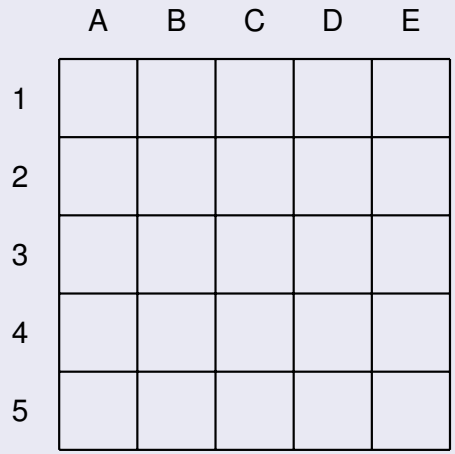
Introduction

General Idea

- Don't search on whole grid
- Use subgoal graph:
  - smaller
  - Nodes = Certain cells which are "subgoals"
  - Edges in the subgoal graph connect nodes which are visible to each other
  - preserve optimality

Introduction

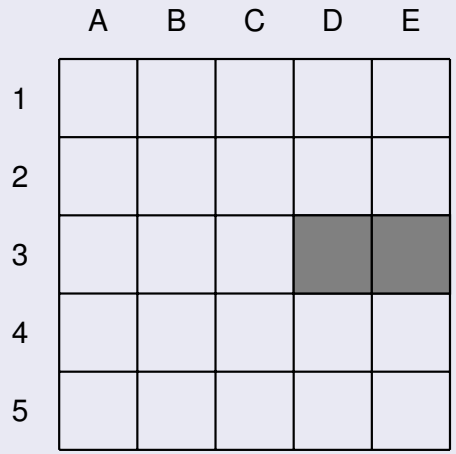
Setting



- 8-neighbor grid

Introduction

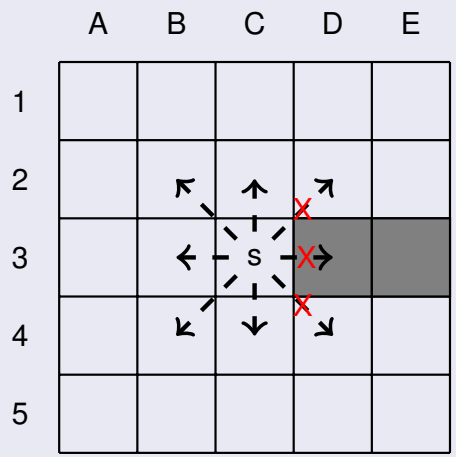
Setting



- 8-neighbor grid
- blocked cells

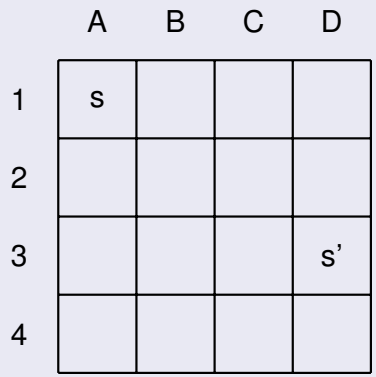
Introduction

Setting



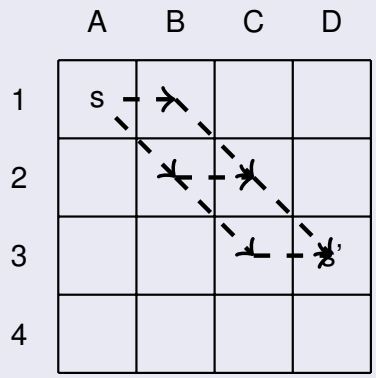
- 8-neighbor grid
- blocked cells
- actor:
  - occupies cell
  - may move in diagonal or cardinal direction if unblocked

Octile Distance



- Used as a heuristic for distance between two cells $h(s, s')$.
- $h(s, s') =$ length of shortest path between s and s', disregarding obstacles

Octile Distance

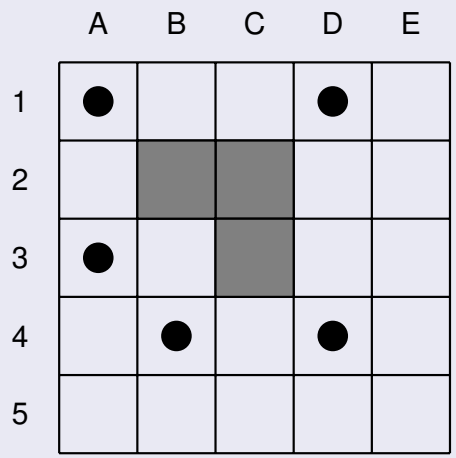


- Used as a heuristic for distance between two cells $h(s, s')$.
- $h(s, s') =$ length of shortest path between s and s', disregarding obstacles
- Here: $h(s, s') = 2\sqrt{2} + 1$

Simple Subgoal Graphs

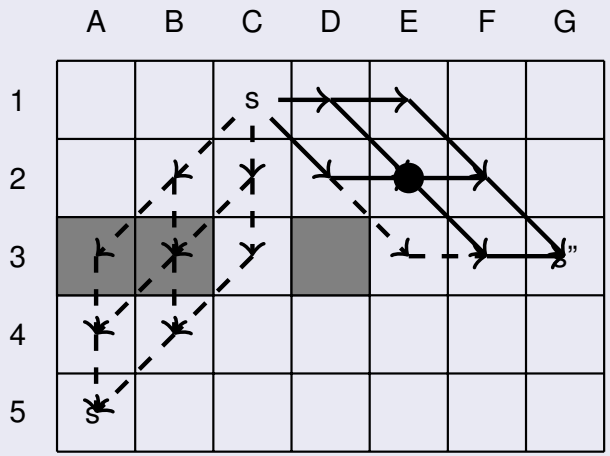
Subgoal

Subgoal



- Subgoals are defined as the cells at corners of obstacles

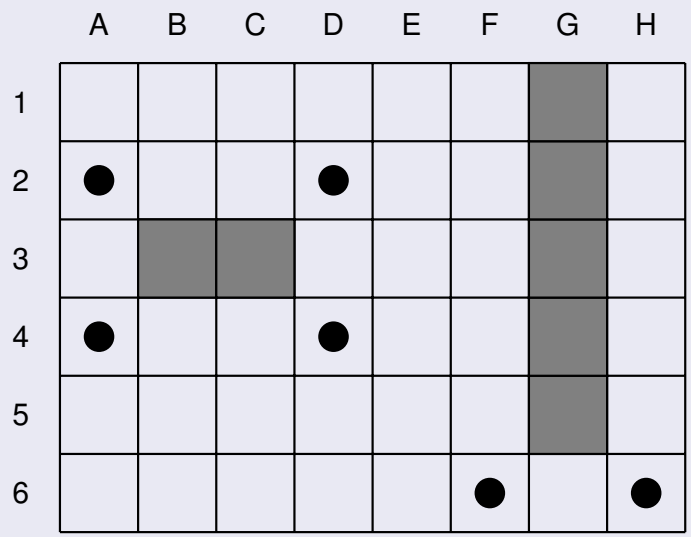
(direct-)h-reachability



Let s and s' be two cells. Cells s and s' are:

- *h-reachable* if there is a shortest path between them
- *direct-h-reachable* if none of the shortest paths between them move over a subgoal

Construction of a Simple Subgoal Graph

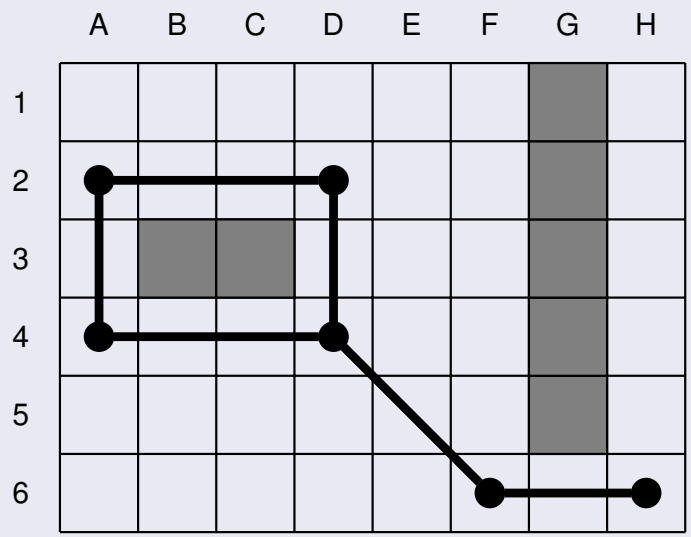


Given a grid:

1. Find subgoals
2. Connect direct-h-reachable subgoals

Grid setting taken from Uras and Koenig [2015]

Construction of a Simple Subgoal Graph

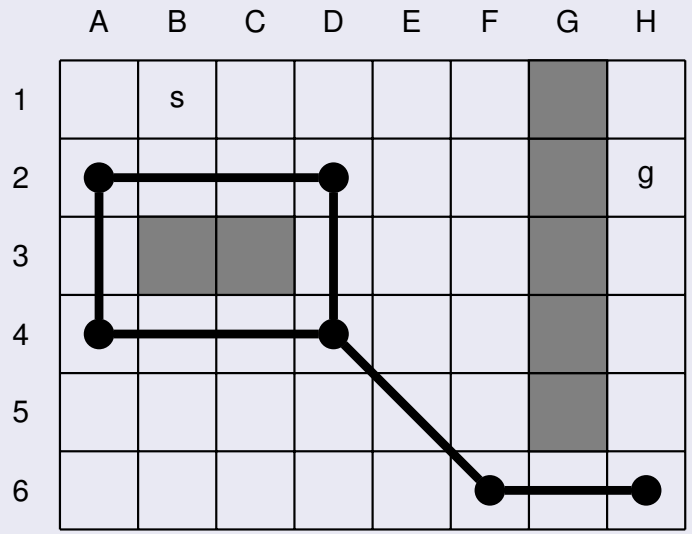


Given a grid:

1. Find subgoals
2. Connect direct-h-reachable subgoals

Grid setting taken from Uras and Koenig [2015]

Searching on a simple subgoal graph

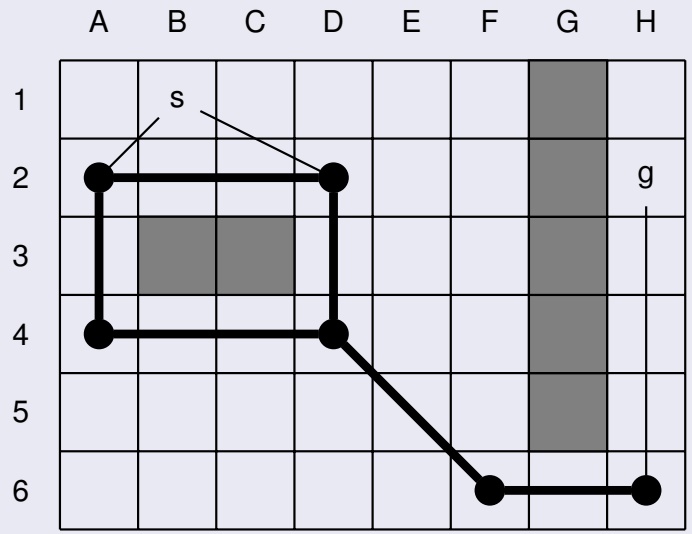


Grid setting taken from Uras and Koenig [2015]

Given a start cell s and a goal cell g:

1. Connect s and g to simple subgoal graph via direct-h-reachable subgoals
2. Search high-level path on simple subgoal graph with A^*
3. Search segments of high-level path for with A^* to find low-level path on grid

Searching on a simple subgoal graph

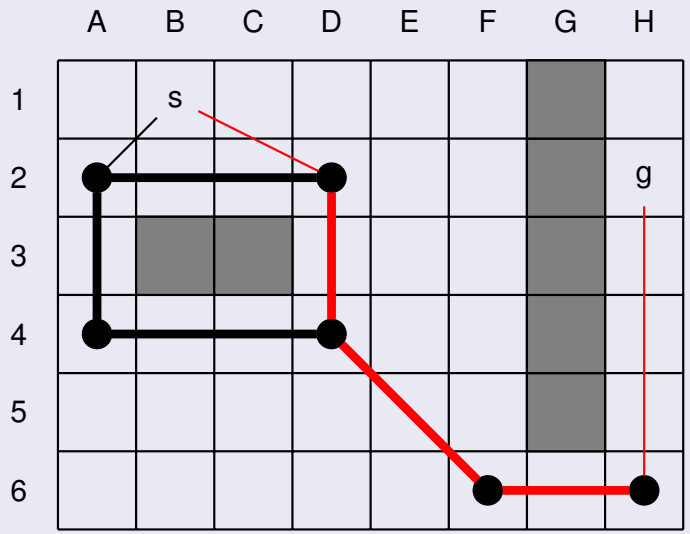


Grid setting taken from Uras and Koenig [2015]

Given a start cell s and a goal cell g:

1. Connect s and g to simple subgoal graph via direct-h-reachable subgoals
2. Search high-level path on simple subgoal graph with A^*
3. Search segments of high-level path for with A^* to find low-level path on grid

Searching on a simple subgoal graph



Grid setting taken from Uras and Koenig [2015]

Given a start cell s and a goal cell g:

1. Connect s and g to simple subgoal graph via direct-h-reachable subgoals
2. Search high-level path on simple subgoal graph with A^*
3. Search segments of high-level path for with A^* to find low-level path on grid

To construct simple subgoal graph, and to connect start and goal cells to simple subgoal graph:

- Need to find all direct-h-reachable subgoals in vicinity of given cell

Speed up process?

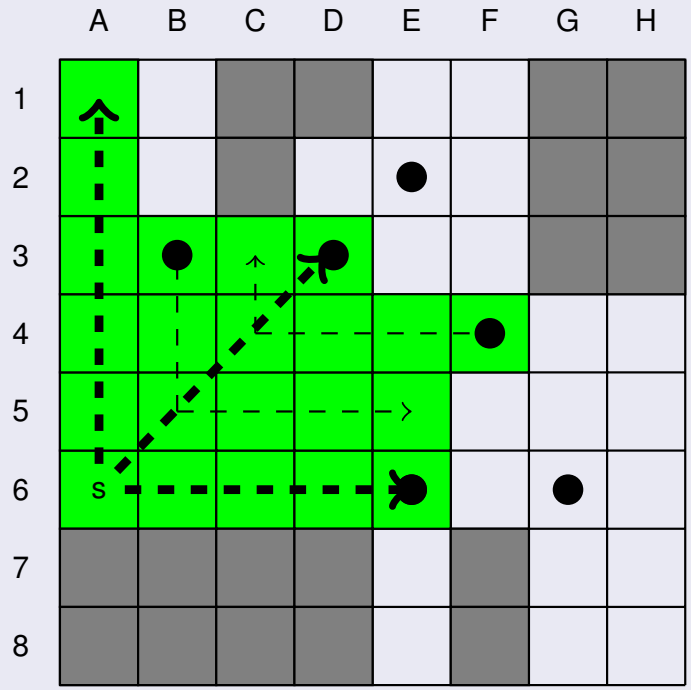
Clearance value

Use clearance values!

Given cell s and cardinal or diagonal direction d:

- $Clearance(s, d) =$ how many moves can we take in direction d
  - i before we reach an obstacle, or
  - ii until we reach a subgoal

Getting all direct-h-reachable subgoals of a given cell



Example taken from Uras et al. [2013]

- Precompute clearance values of every cell on the grid in all diagonal and cardinal directions

Then, given cell *s*:

1. Partition space around *s* in octants.
2. For each octant, sweep horizontal lines beginning from diagonal lines

Two-Level Subgoal Graphs

Idea

Decrease search space with further abstraction:

- Partition subgoals into local and global subgoals, prune local subgoals from graph
- Perform high-level search only on global subgoals

Local and global subgoals

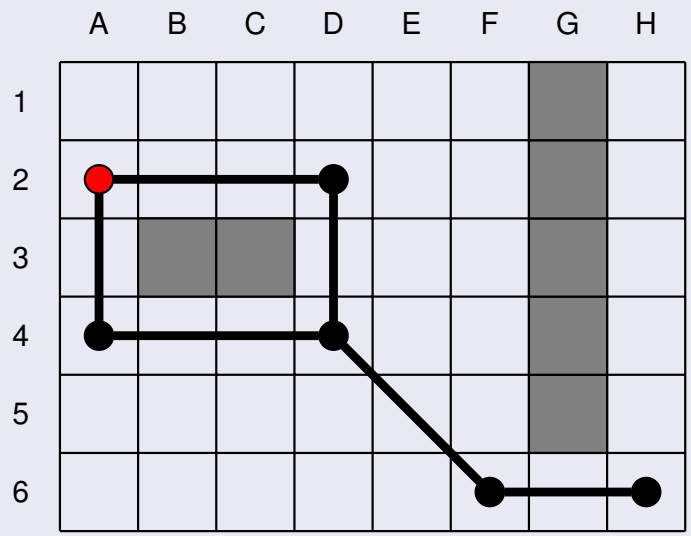
Given subgoal s and subgoals s' and s'' to which it is connected via an edge.
s is a local subgoal if:

i. there exists path from s' to s'' through only global subgoals and not through s.
   - May not be longer than the original path through s!

   or

ii. s' and s'' are h-reachable

Construction of a Two-Level Subgoal Graph



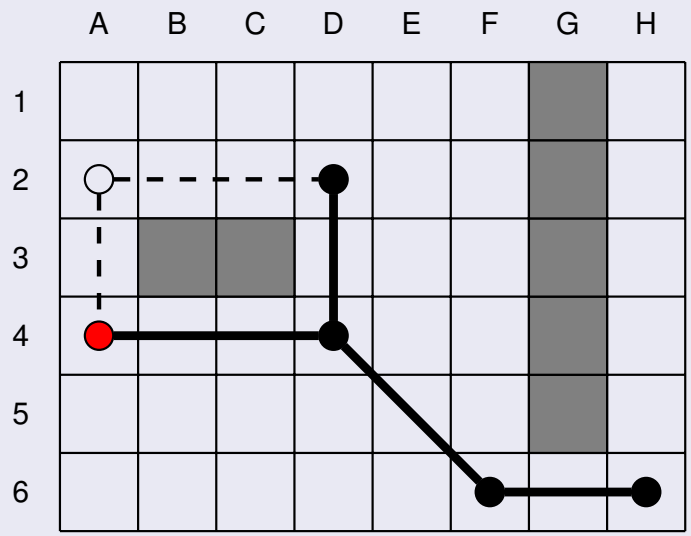
Subgoal *A*2:

- Neighbors *D*2 and *A*4
  - i $\surd$
  - ii $\times$

*A*2 is local!

Grid setting from Uras and Koenig [2015]

Construction of a Two-Level Subgoal Graph



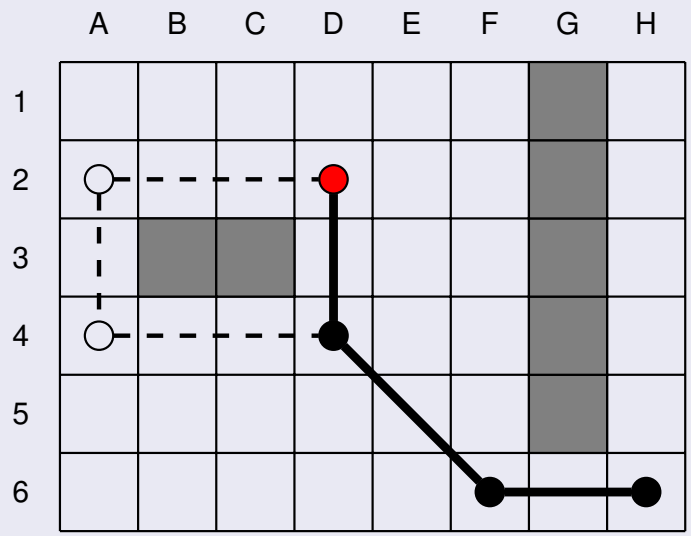
Subgoal *A*4:

- Neighbors *A*2 and *D*4
  - i $\surd$
  - ii $\times$

*A*4 is local!

Grid setting from Uras and Koenig [2015]

Construction of a Two-Level Subgoal Graph



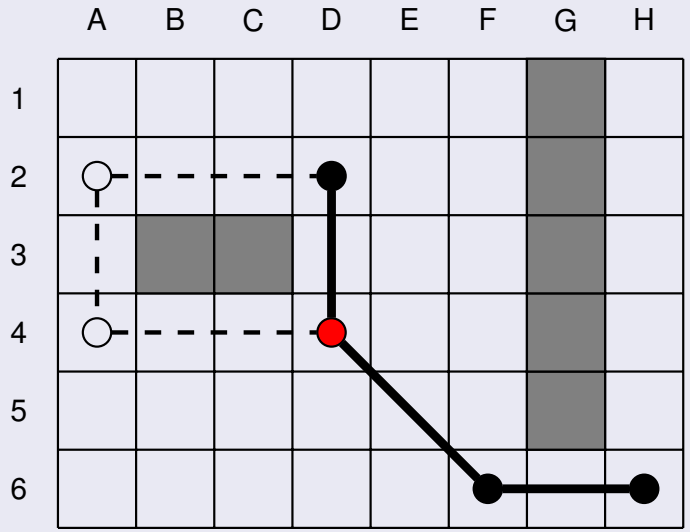
Subgoal *D*2:

- Neighbors *A*2 and *D*4
  - i ×
  - ii ×

*D*2 is global!

Grid setting from Uras and Koenig [2015]

Construction of a Two-Level Subgoal Graph



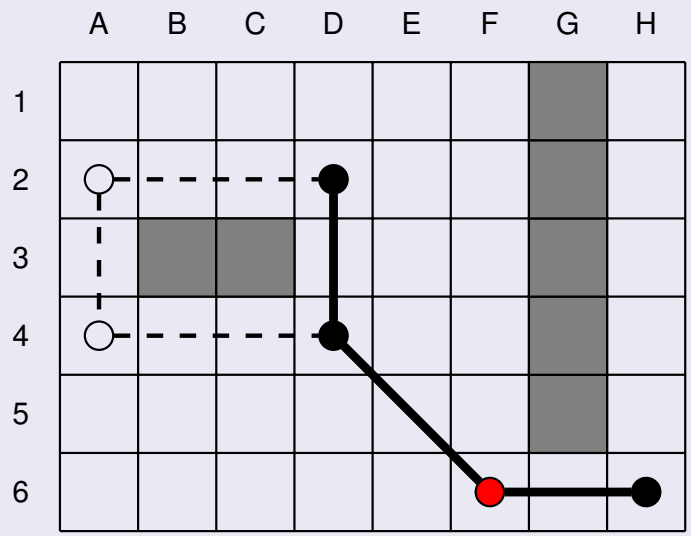
Grid setting from Uras and Koenig [2015]

Subgoal *D*4:

- Neighbors *D*2 and *A*4
  - i ×
  - ii ×
- Neighbors *A*4 and *F*6
  - i ×
  - ii √
- Neighbors *D*2 and *F*6
  - i ×
  - ii √

*D*4 is global!

Construction of a Two-Level Subgoal Graph



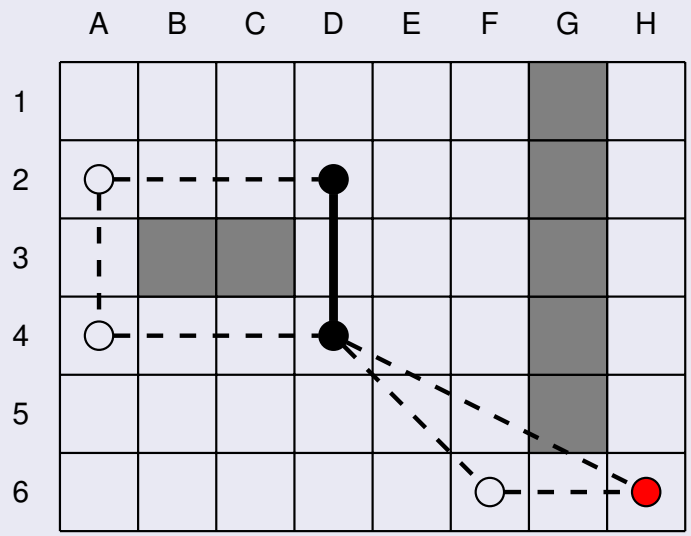
Subgoal *F*6:

- Neighbors *D*4 and *H*6
  - i ×
  - ii √

Add edge between *D*4 and *H*6, *F*6 is then local

Grid setting from Uras and Koenig [2015]

Construction of a Two-Level Subgoal Graph



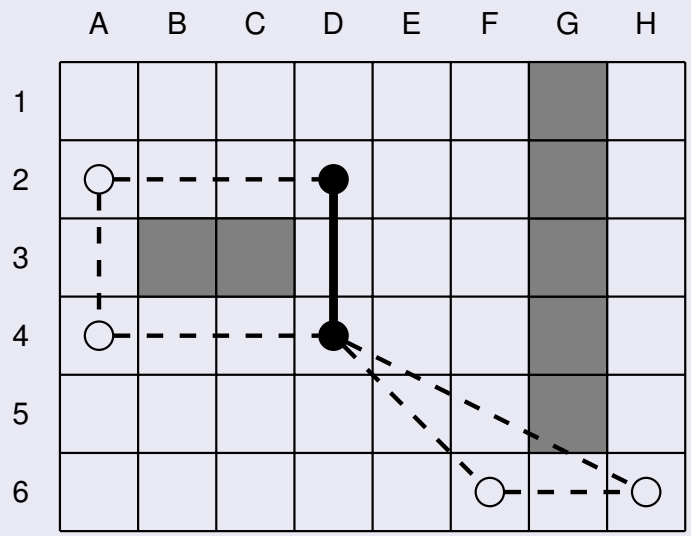
Subgoal *H*6:

- Neighbors *D*4 and *F*6
  - i √
  - ii √

H6 is local!

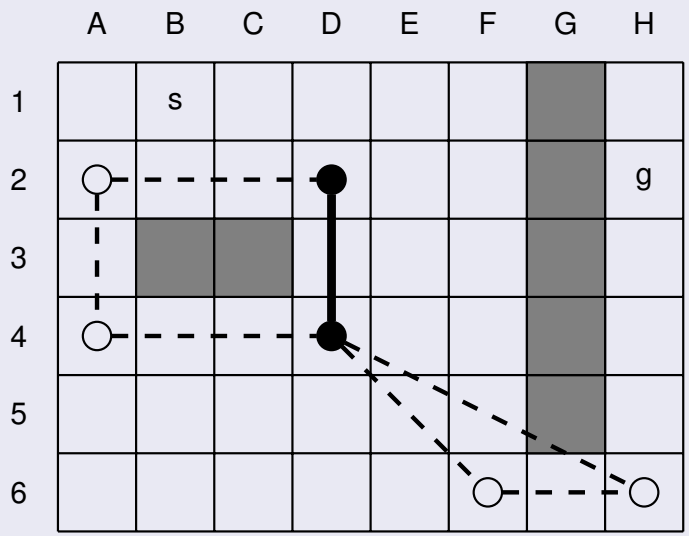
Grid setting from Uras and Koenig [2015]

Construction of a Two-Level Subgoal Graph



Grid setting from Uras and Koenig [2015]

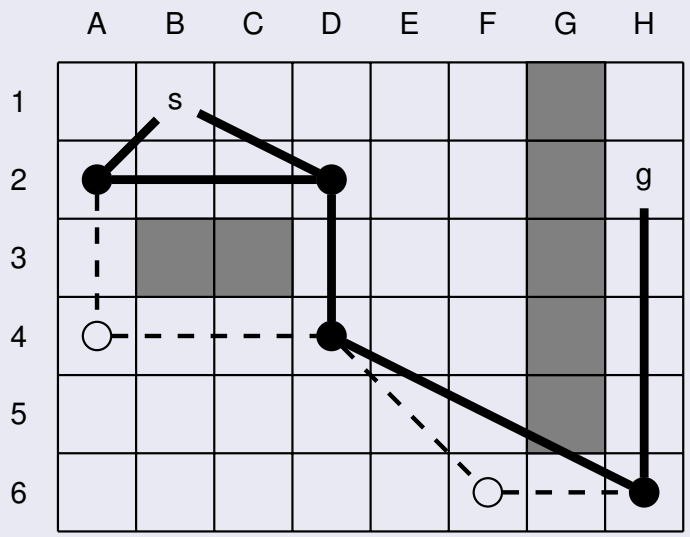
Searching in a Two-Level Subgoal Graph



Grid setting from Uras and Koenig [2015]

1. Connect *s* and *g* to global subgoals.
   - direct-h-reachable!
   - may need to make some local subgoals global, temporarily
2. search for high-level path on global subgoals
3. search for actual path on grid between each segment

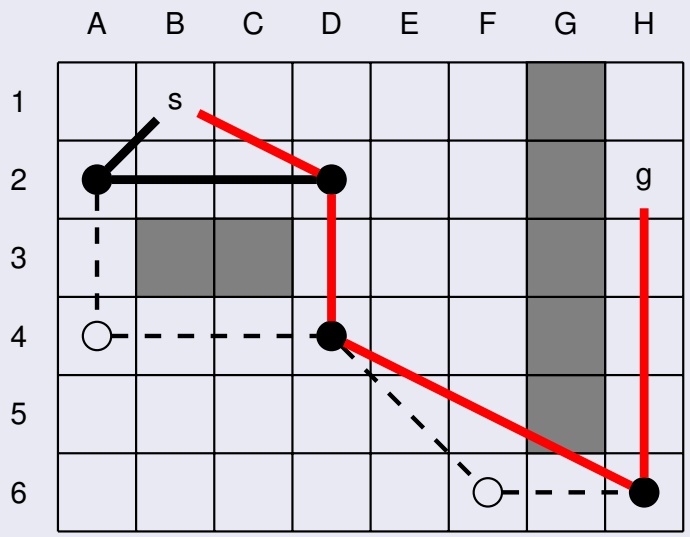
Searching in a Two-Level Subgoal Graph



Grid setting from Uras and Koenig [2015]

1. Connect *s* and *g* to global subgoals.
   - direct-h-reachable!
   - may need to make some local subgoals global, temporarily
2. search for high-level path on global subgoals
3. search for actual path on grid between each segment

Searching in a Two-Level Subgoal Graph



Grid setting from Uras and Koenig [2015]

1. Connect *s* and *g* to global subgoals.
   - direct-h-reachable!
   - may need to make some local subgoals global, temporarily
2. search for high-level path on global subgoals
3. search for actual path on grid between each segment

Further remarks

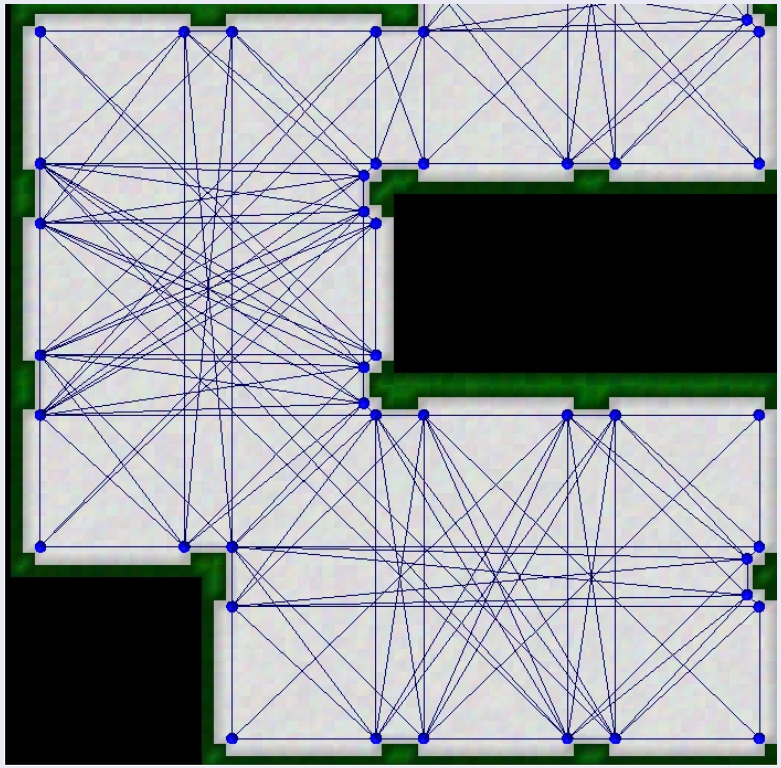


Figure: Single Subgoal Graph [Uras et al., 2013]

Two-Level subgoal graphs may drastically reduce search space!

- Especially on bigger maps

But:

- More complex search between subgoals

Further remarks

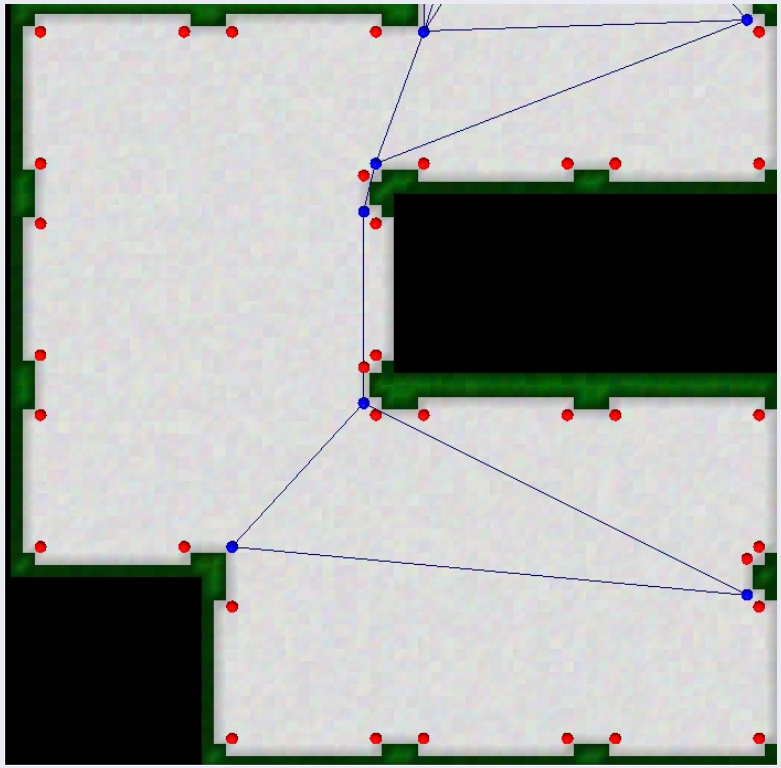


Figure: Two-Level Subgoal Graph [Uras et al., 2013]

Two-Level subgoal graphs may drastically reduce search space!

- Especially on bigger maps

But:

- More complex search between subgoals

Conclusion

Conclusion

Simple Subgoal graphs

- preprocessing strategy for 8-neighbor grid problems
- Reduce search space
  - search nodes = corners of obstacles, “subgoals”
  - edges between direct-h-reachable subgoals

Two-Level subgoal graphs

- Further abstraction to reduce search space
- Prune graph and retain only global nodes: h-reachable!
- Strong performance (Grid-Based Path Planning Competition 2012 and 2013 “nondominated”)

Thank you for your attention!

References

Tansel Uras and Sven Koenig. Subgoal graphs for fast optimal pathfinding. In Steven Rabin, editor, *Game AI Pro 2: Collected Wisdom of Game AI Professionals*. A K Peters/CRC Press, 2015.

Tansel Uras, Sven Koenig, and Carlos Hernández. Subgoal graphs for optimal pathfinding in eight-neighbor grids. In *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling, ICAPS 2013, Rome, Italy, June 10-14, 2013*, 2013. URL `http://www.aaai.org/ocs/index.php/ICAPS/ICAPS13/paper/view/6058`.