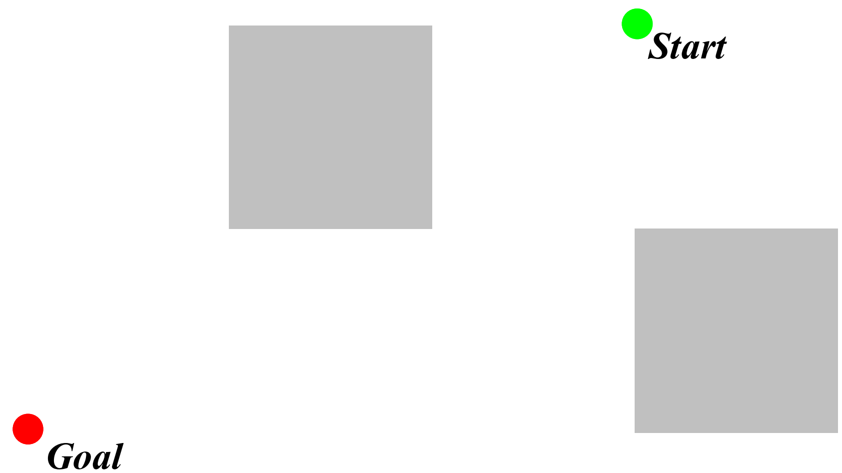
Any-Angle Search

- Do not restrict paths to a grid or an a-priori given graph to find optimal paths in continuous environments

Any-Angle Search

[from JPL]



- A. Nash and S. Koenig. Any-Angle Path Planning. Artificial Intelligence Magazine, 34(4), 85-107, 2013.

A* on Visibility Graphs

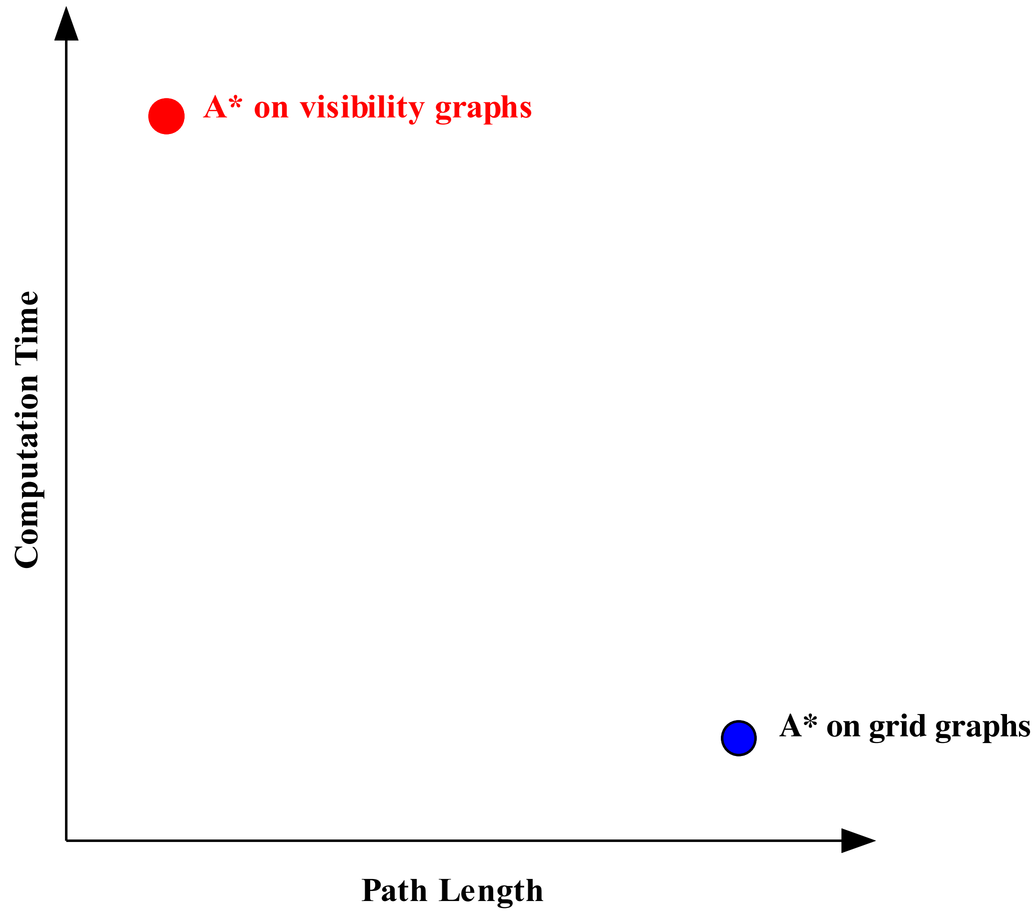
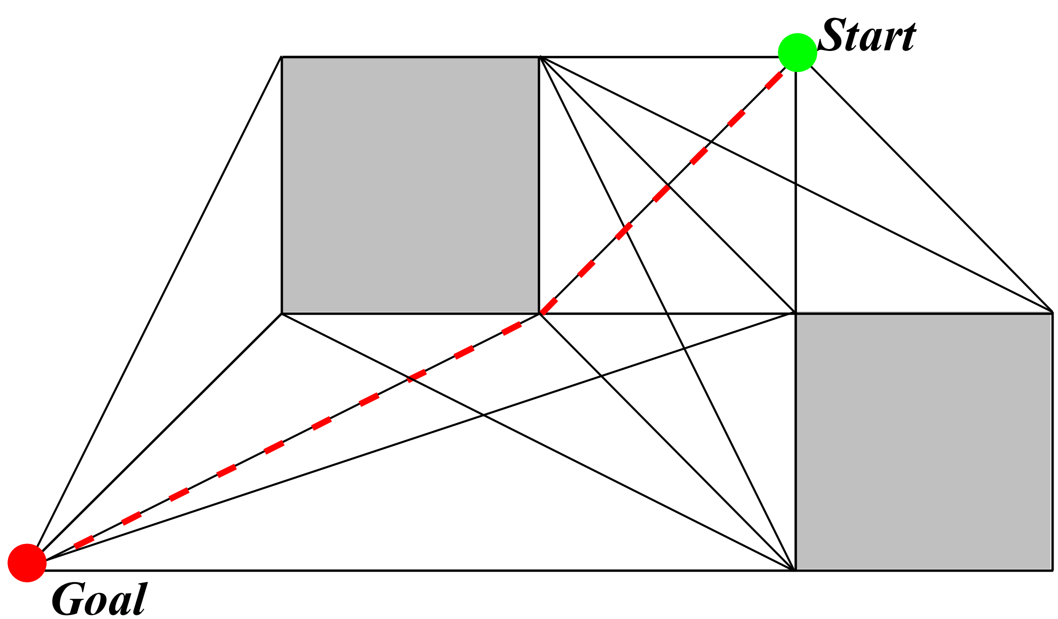


figure is notional

A* on Visibility Graphs

- A* on Visibility Graphs [Lozano-Perez et al.]
- Note: Sophisticated versions exist, e.g. [Shah and Gupta]



- Shortest path in 2D terrain
- Slow due to many edges and line-of-sight checks

A* on Grid Graphs

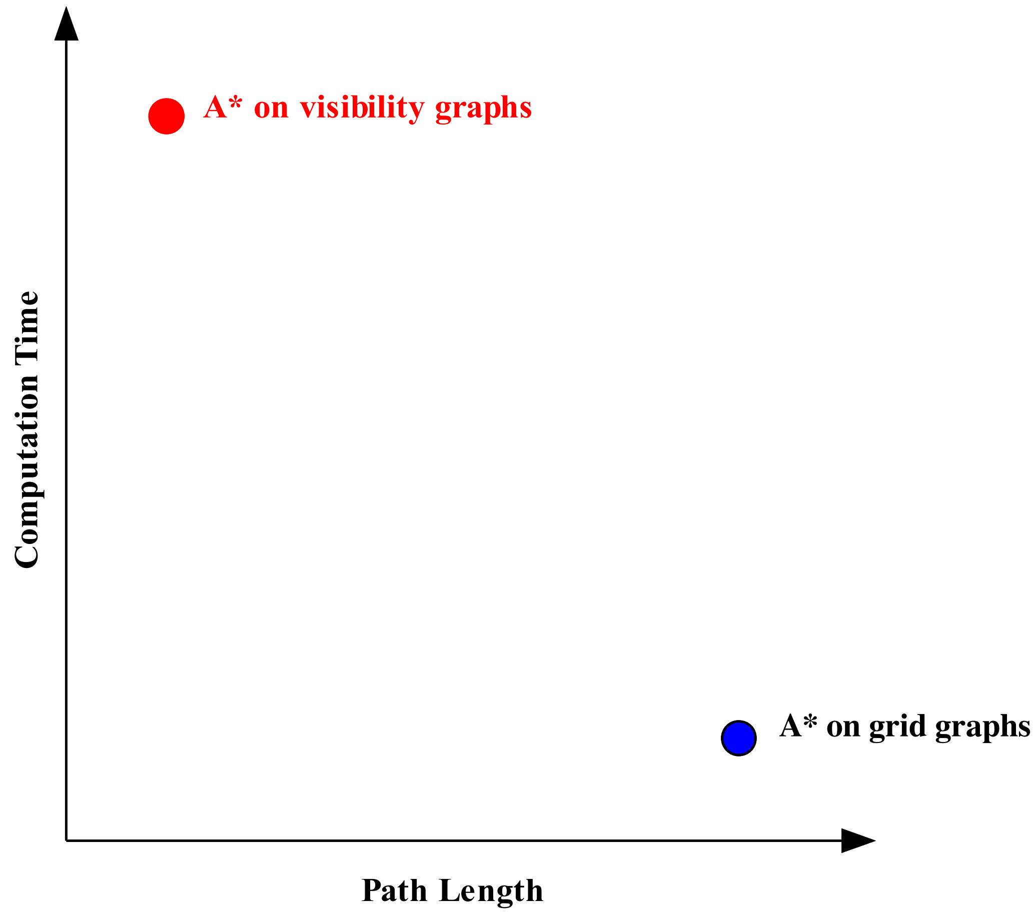
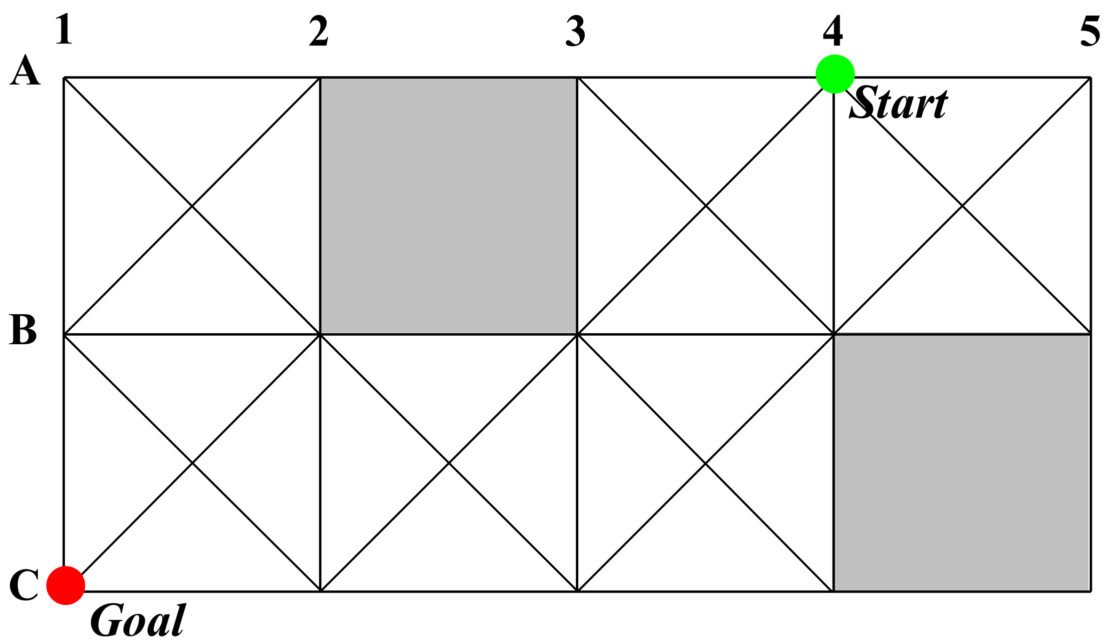


figure is notional

A* on Grid Graphs

- A* on grid graphs

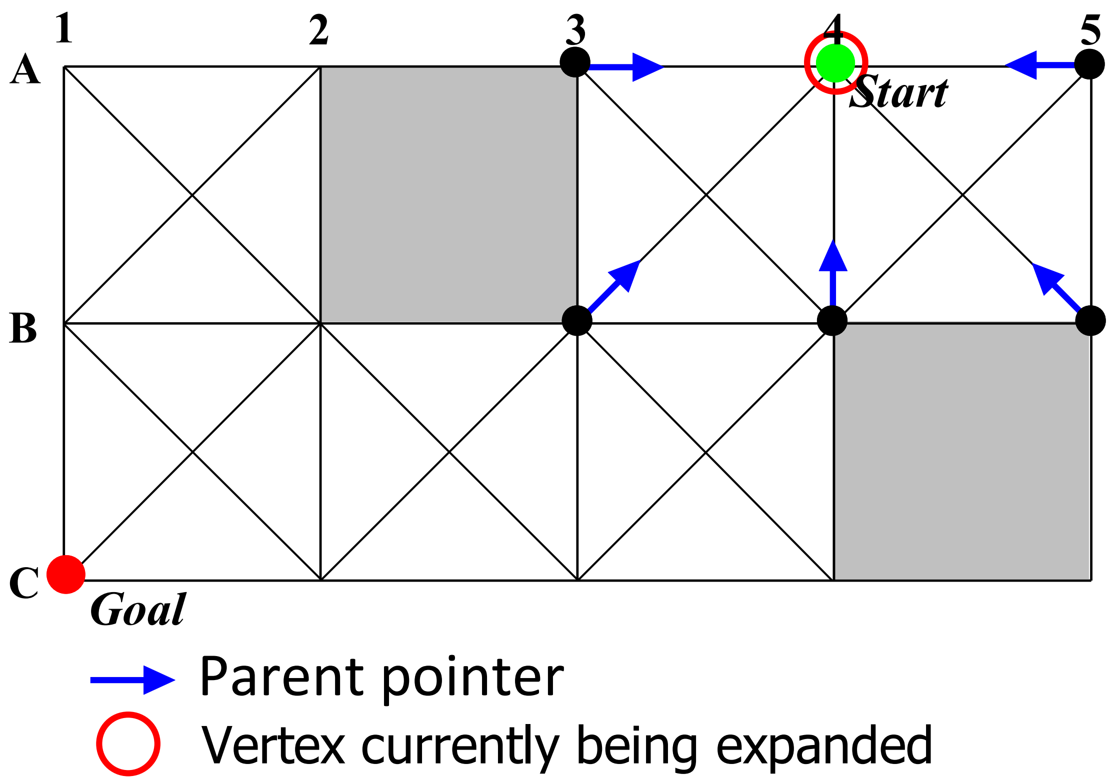


- A* assigns two values to every vertex s
  - g(s): the length of the shortest path from the start vertex to s found so far
  - parent(s): the parent pointer used to extract the path after termination
  - Following the parents from s to the start vertex results in a path of length g(s)

8-neighbor grid

A* on Grid Graphs

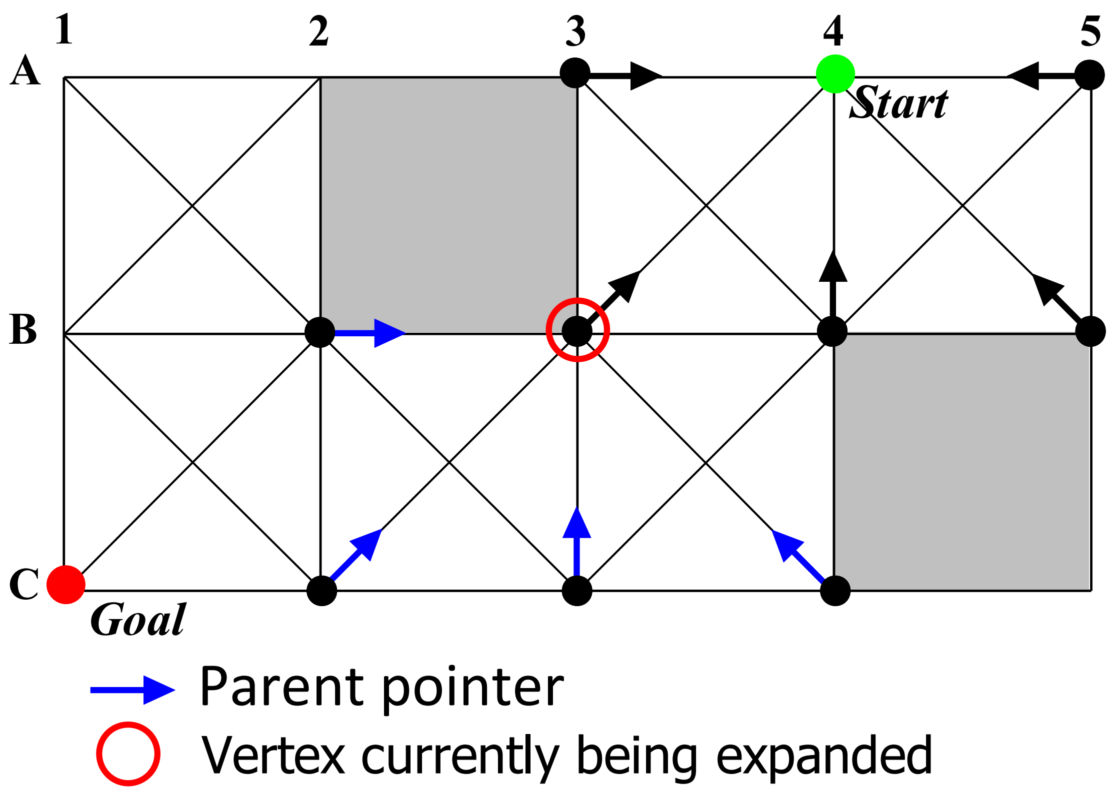
- A* on grid graphs



8-neighbor grid

A* on Grid Graphs

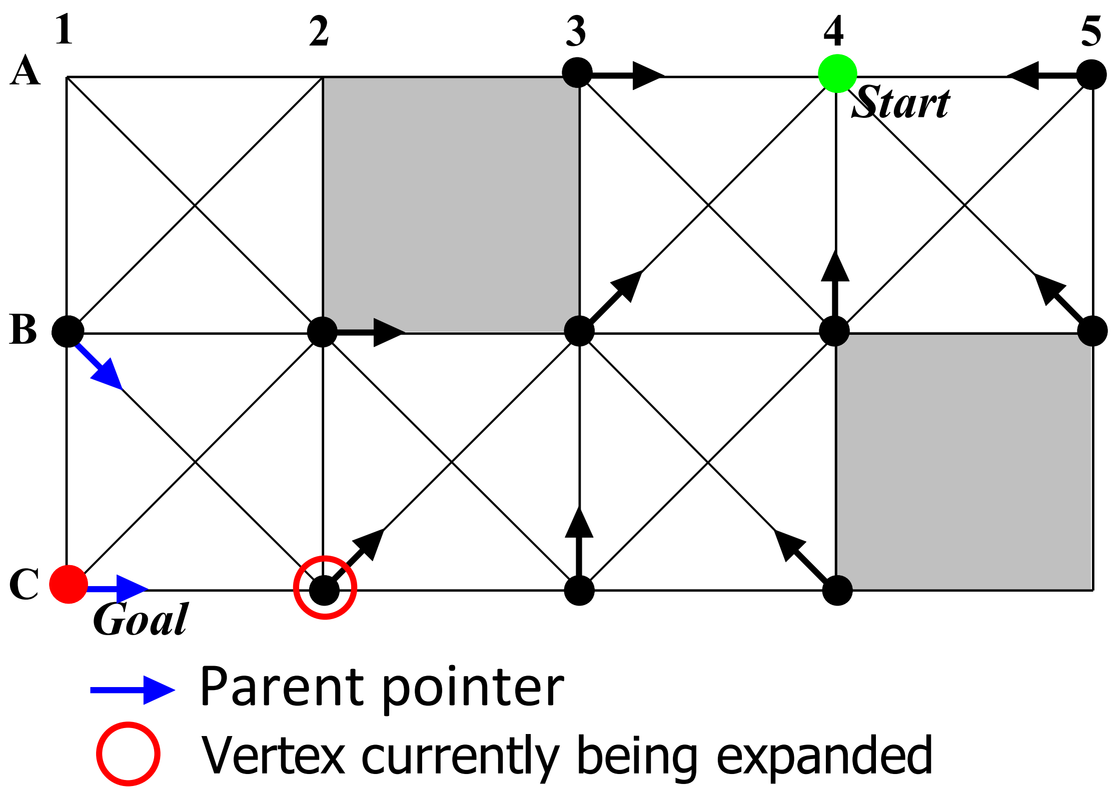
- A* on grid graphs



8-neighbor grid

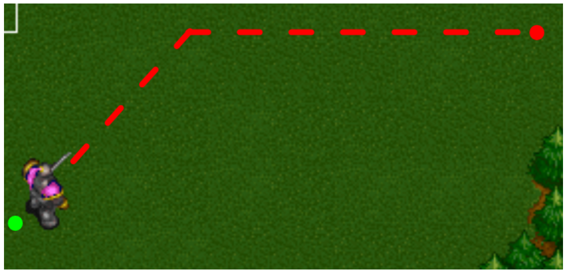
A* on Grid Graphs

- A* on grid graphs

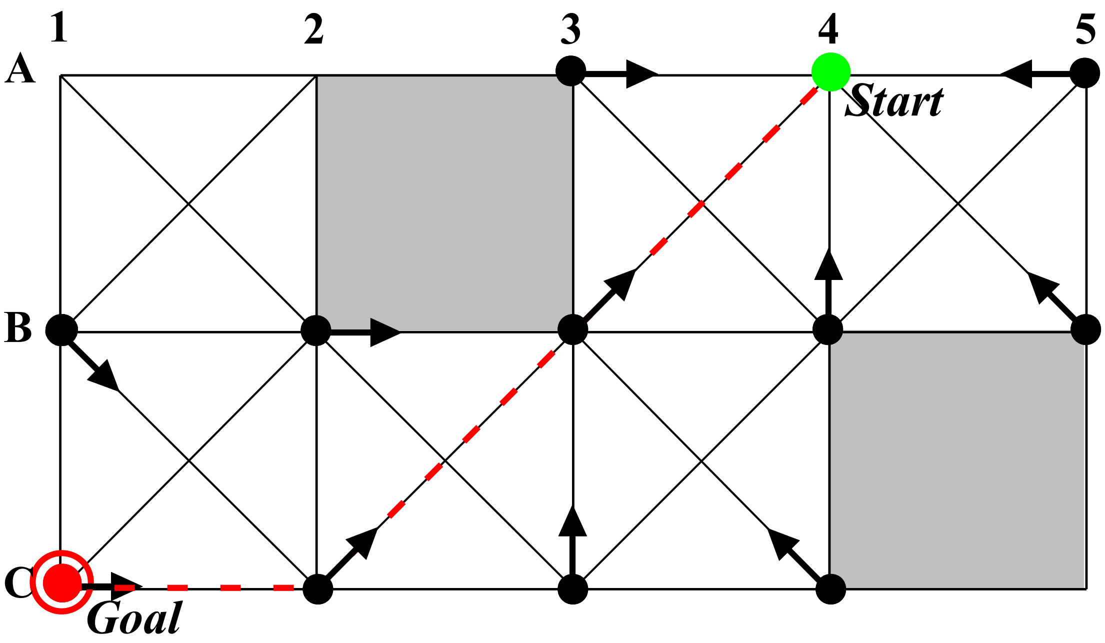


8-neighbor grid

A* on Grid Graphs



- A* on grid graphs



Parent pointer

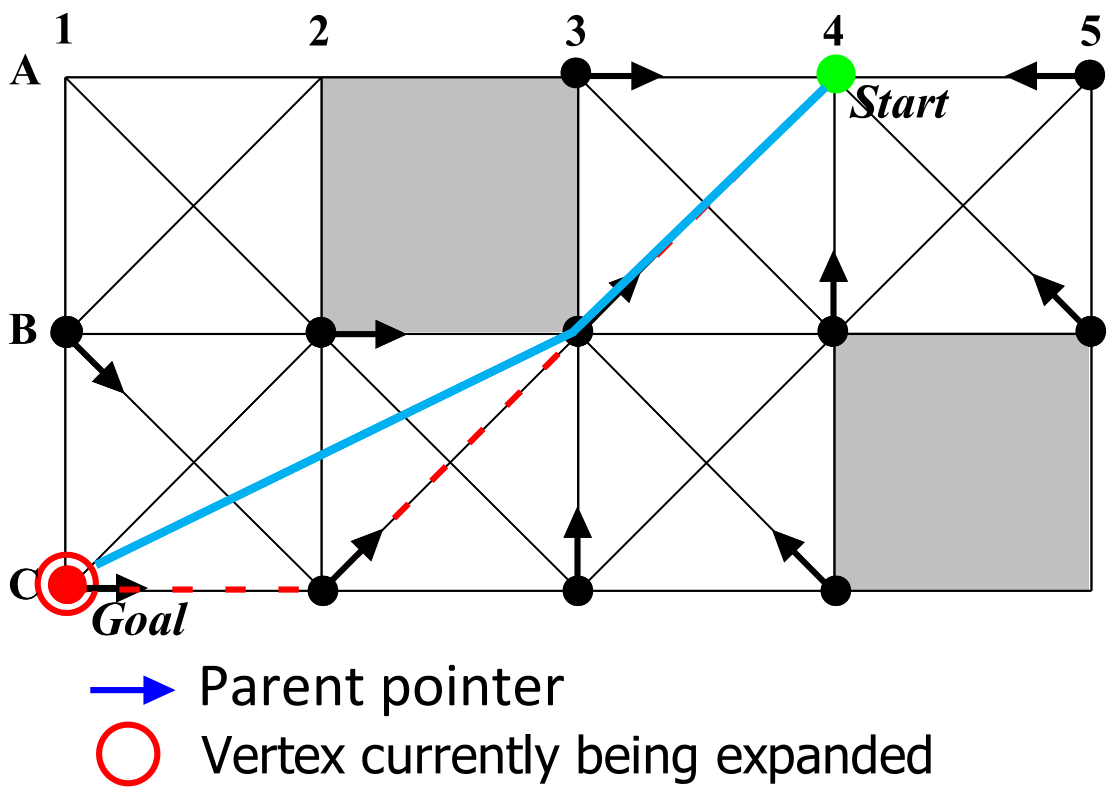
Vertex currently being expanded

8-neighbor grid

from now on, we will show only the cells

A* on Grid Graphs

- A* on grid graphs



8-neighbor grid

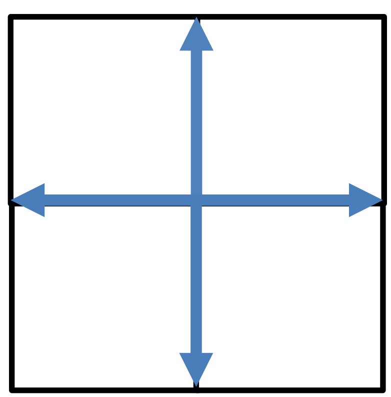
from now on, we will show only the cells

A* on Grid Graphs

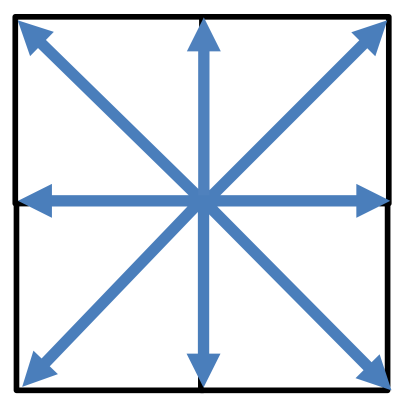
Dimension	Regular Grid	Neighbors	% Longer Than Shortest Path
2D	triangular grid corners	3-neighbor	≈ 100
		6-neighbor	≈ 15
	square grid corners	4-neighbor	≈ 41
		8-neighbor	≈ 8
	hexagonal grid centers	6-neighbor	at least ≈ 15
		12-neighbor	at least ≈ 4
3D	cubic grid corners	6-neighbor	at least ≈ 73
		26-neighbor	at least ≈ 13

Grids with Higher Degree Vertices

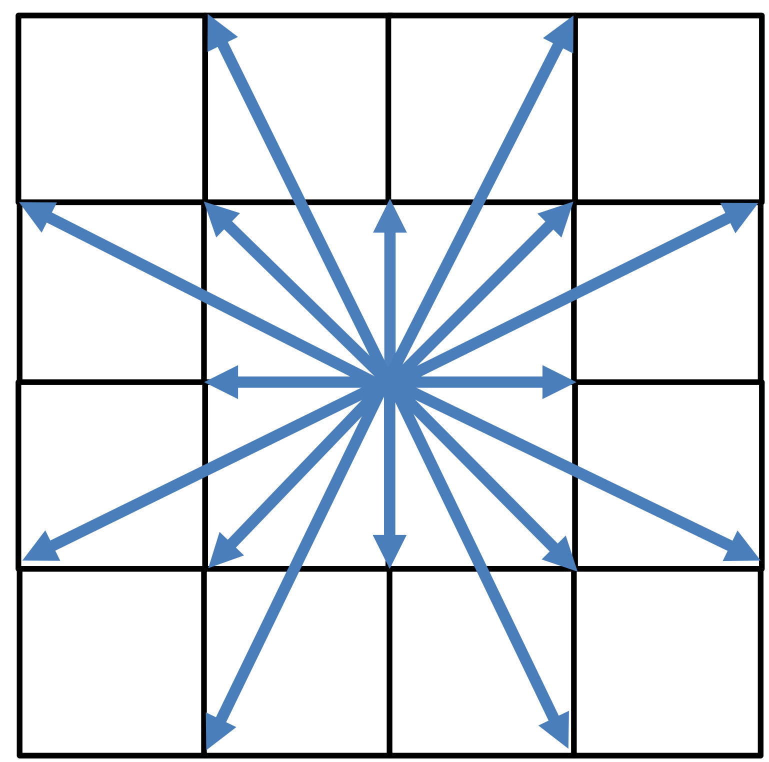
- Grid path finding on the 2^k neighborhoods [Rivera et al.]



$2^2=4$
neighborhood



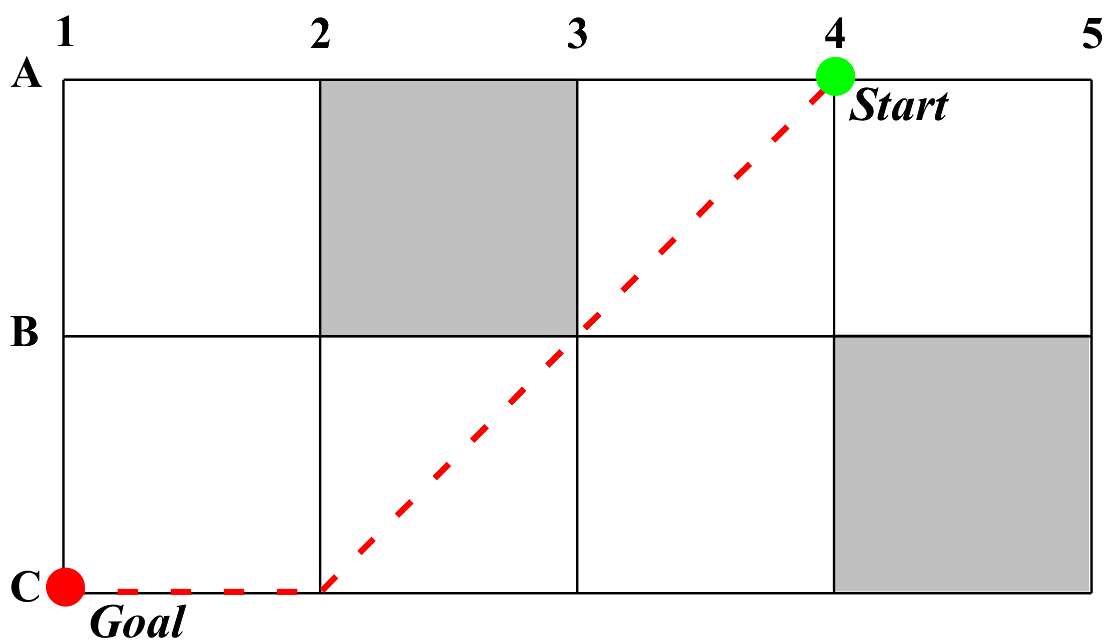
$2^3=8$
neighborhood



$2^4=16$
neighborhood

A* with Post Smoothing

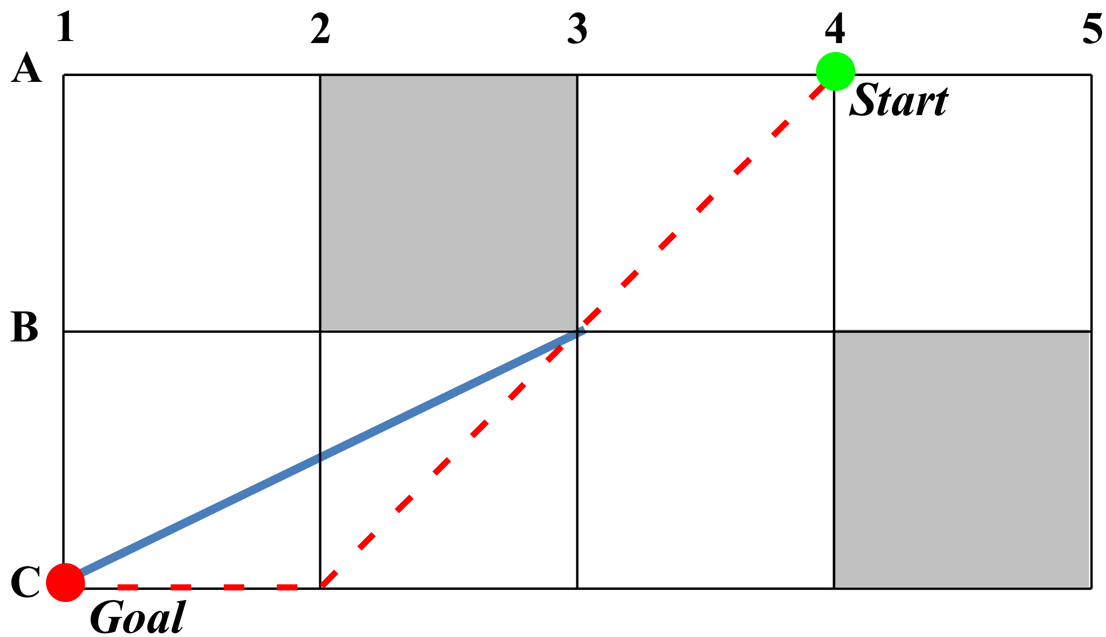
- A* with Post Smoothing [Thorpe; Botea et al.; Millington]



8-neighbor grid

A* with Post Smoothing

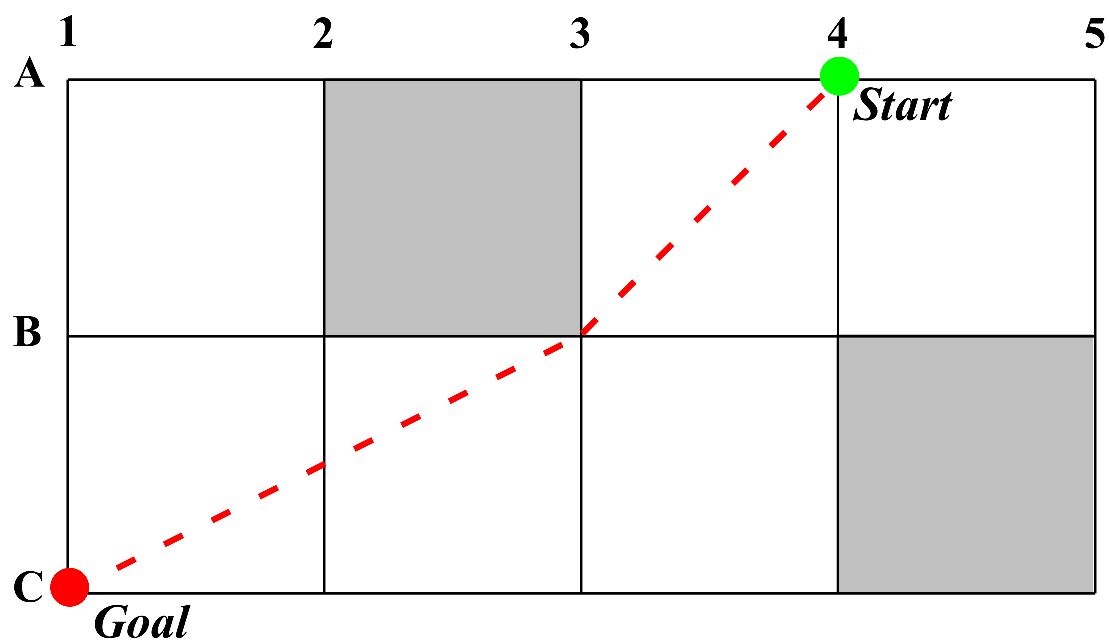
- A* with Post Smoothing



8-neighbor grid

A* with Post Smoothing

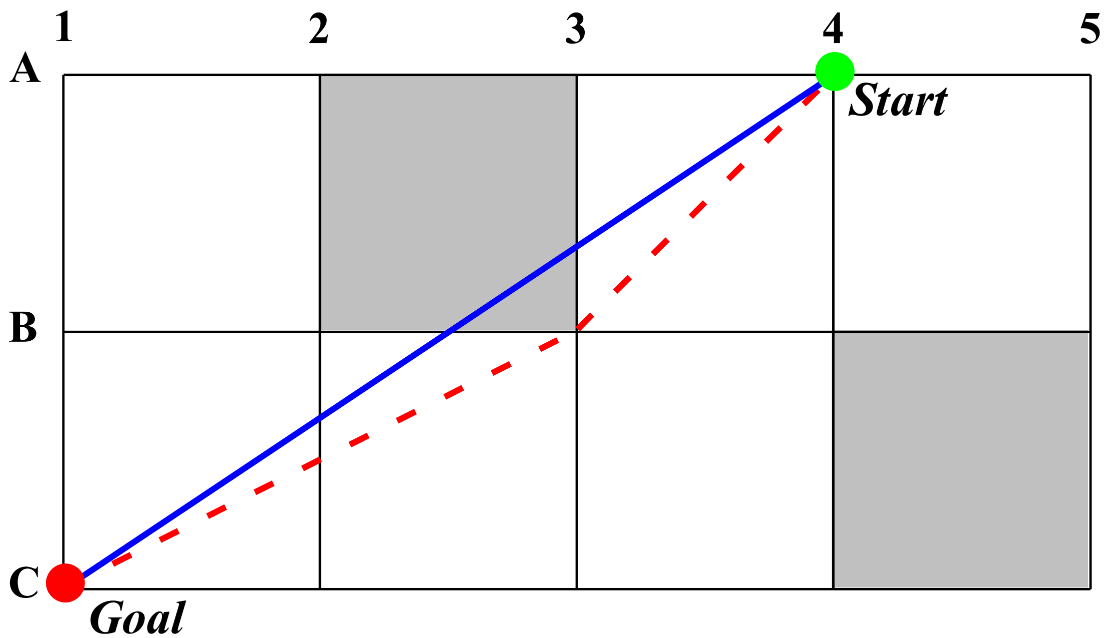
- A* with Post Smoothing



8-neighbor grid

A* with Post Smoothing

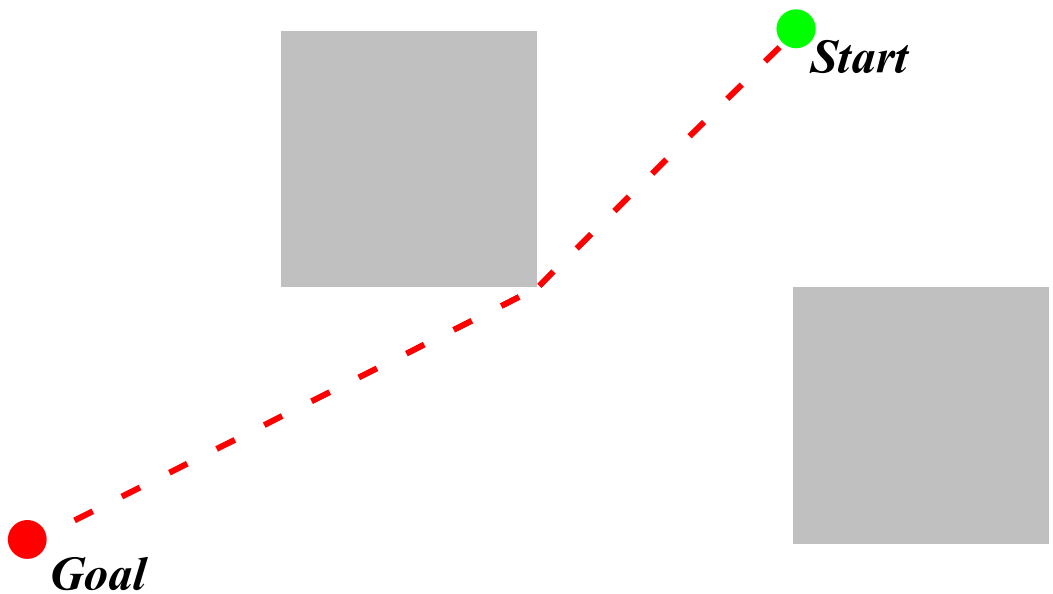
- A* with Post Smoothing



8-neighbor grid

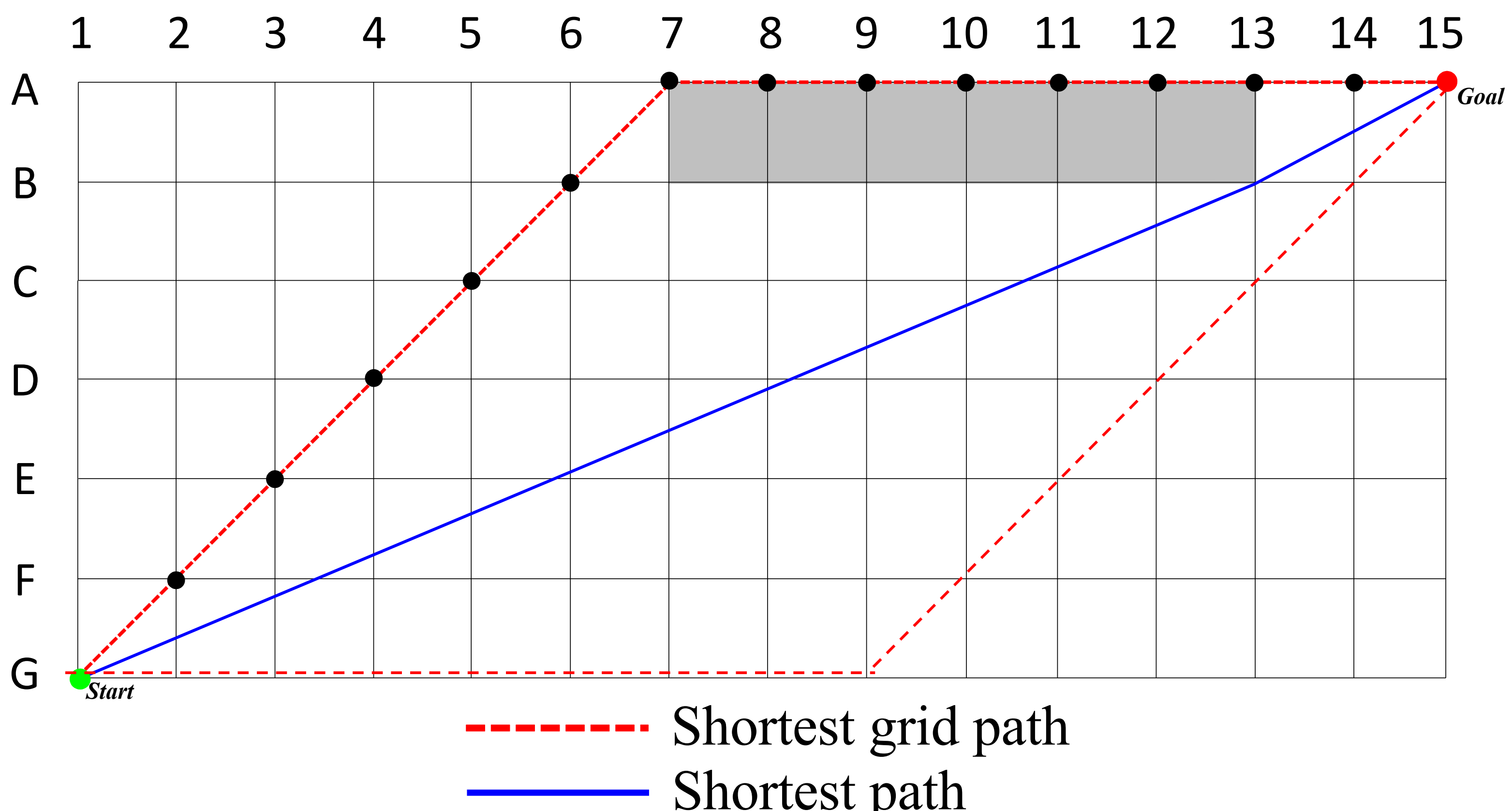
A* with Post Smoothing

- A* with Post Smoothing



A* with Post Smoothing

- A* with Post Smoothing



- Postprocessing often leaves path homotopy unchanged
- Better to interleave the search and the optimization

Suboptimal Any-Angle Search

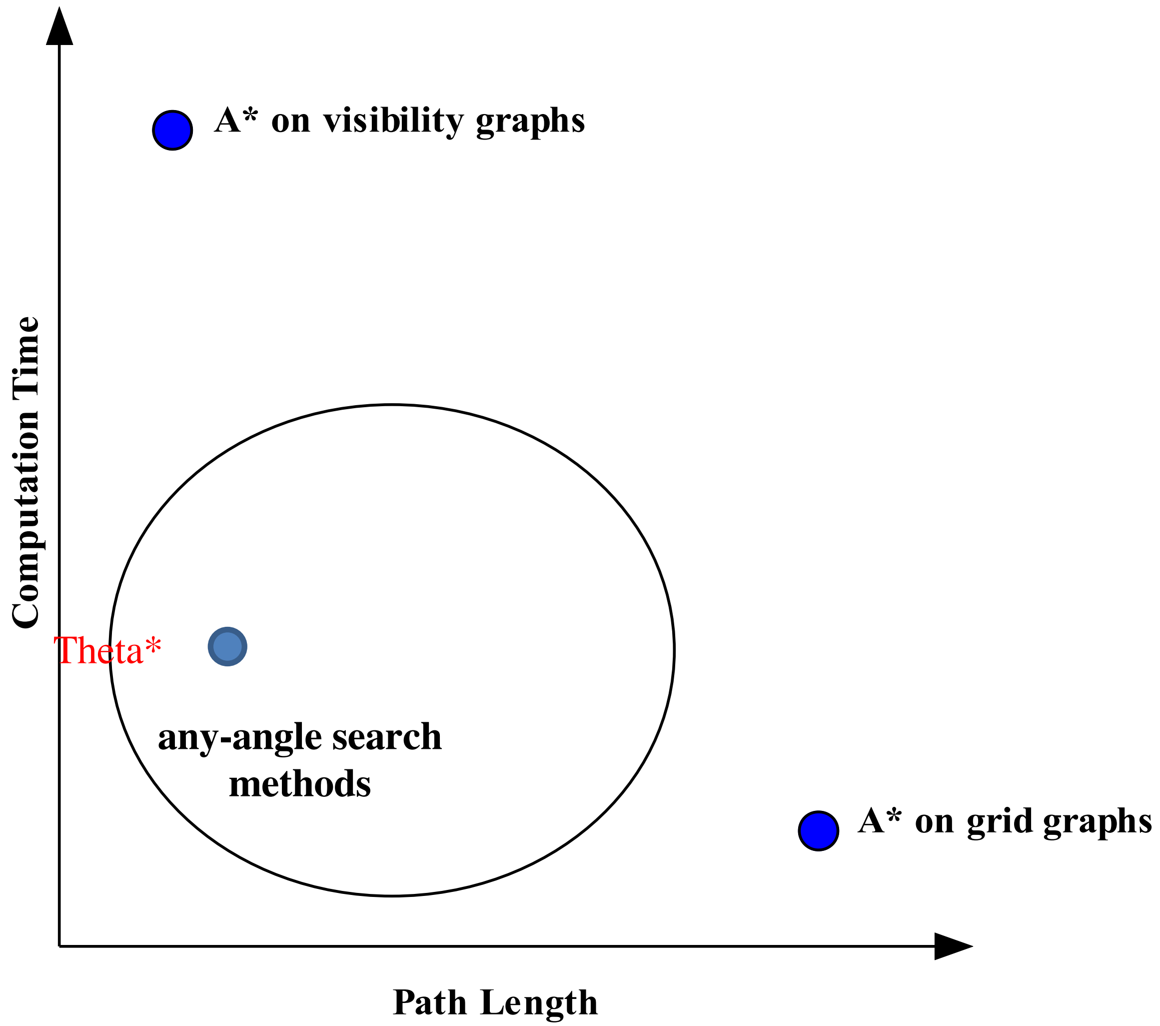
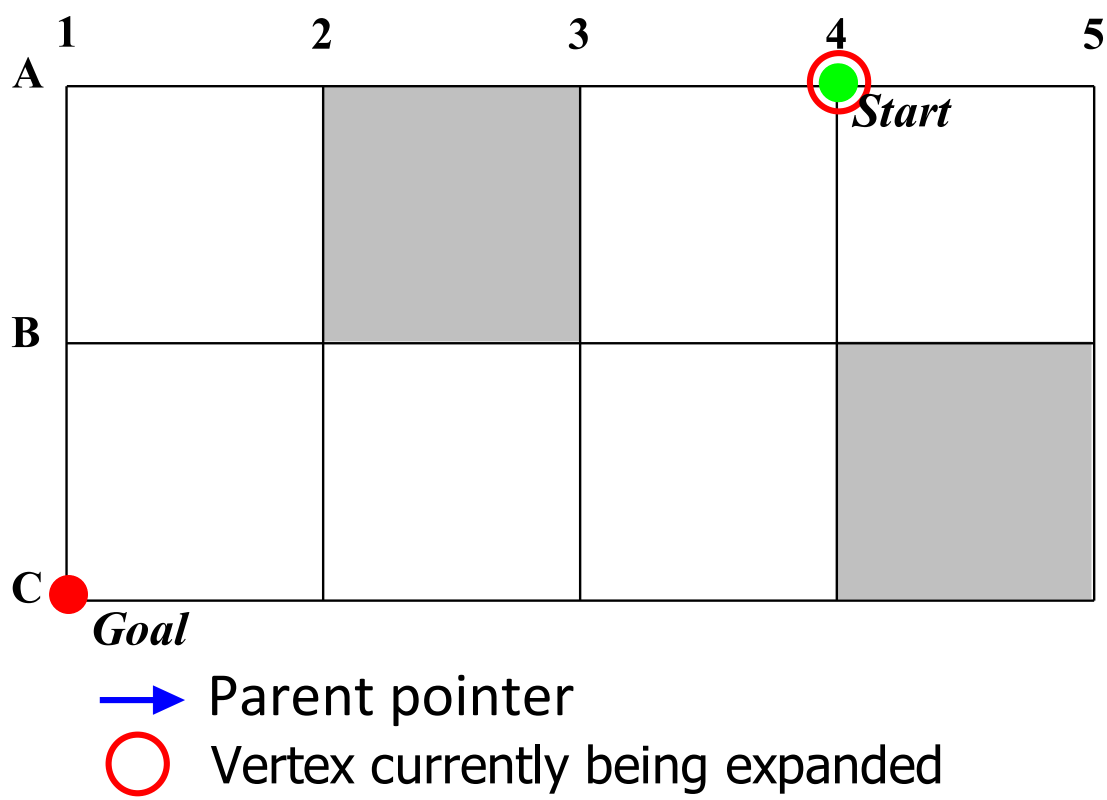


figure is notional

Suboptimal Theta*

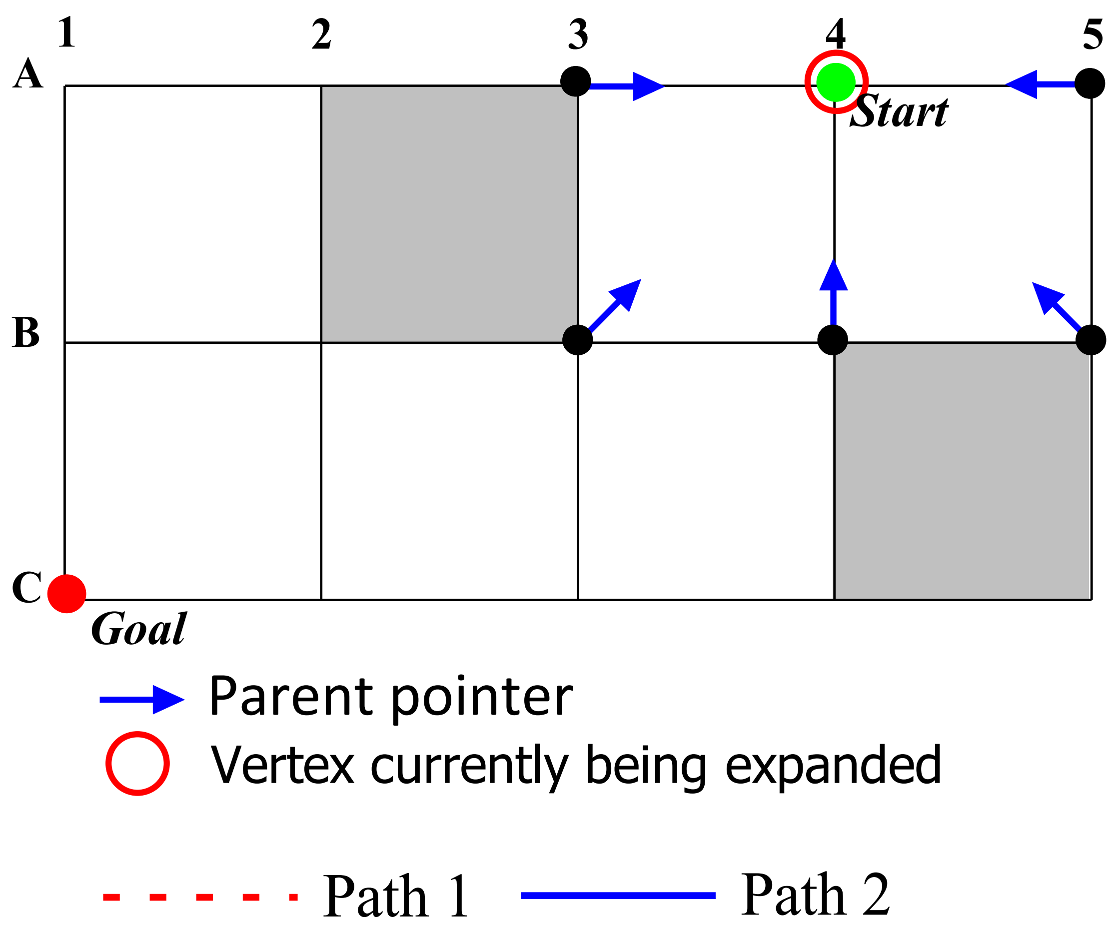
- Theta*



8-neighbor grid

Suboptimal Theta*

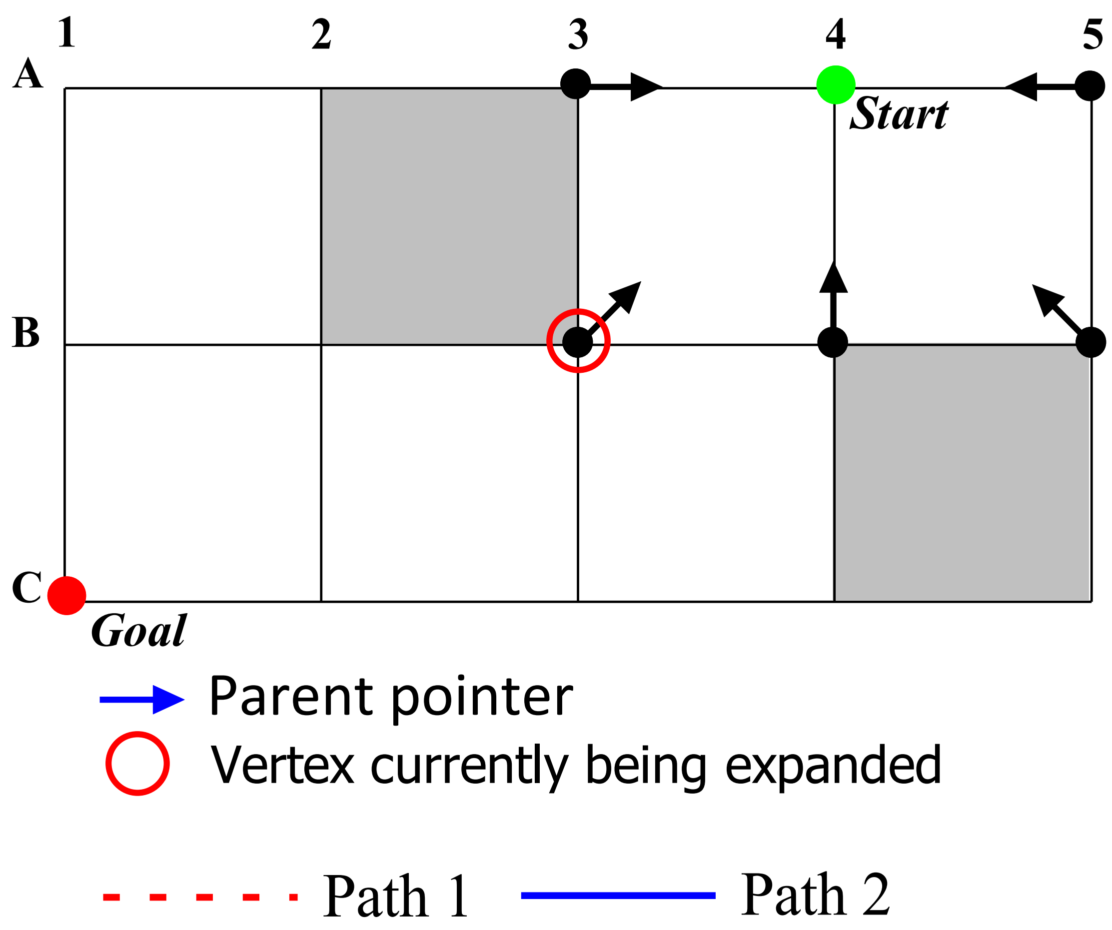
- Theta*



8-neighbor grid

Suboptimal Theta*

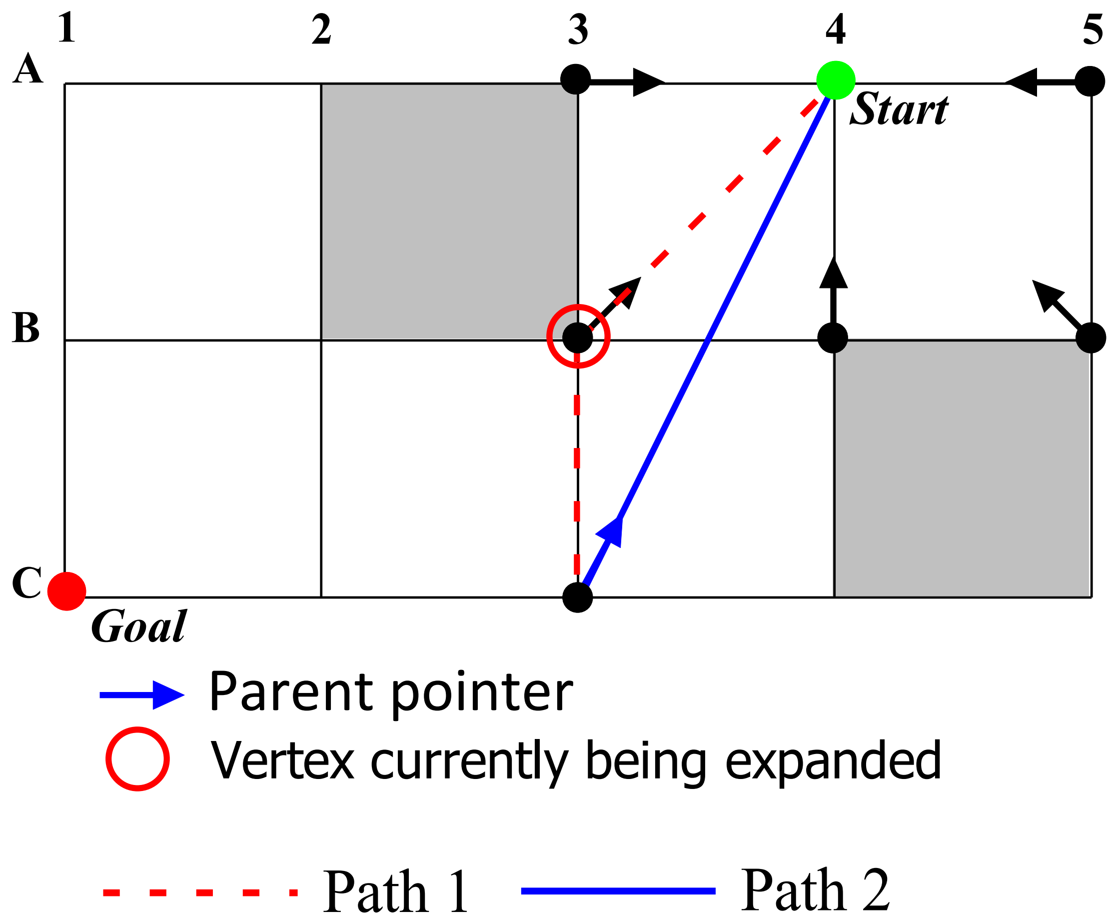
- Theta*



8-neighbor grid

Suboptimal Theta*

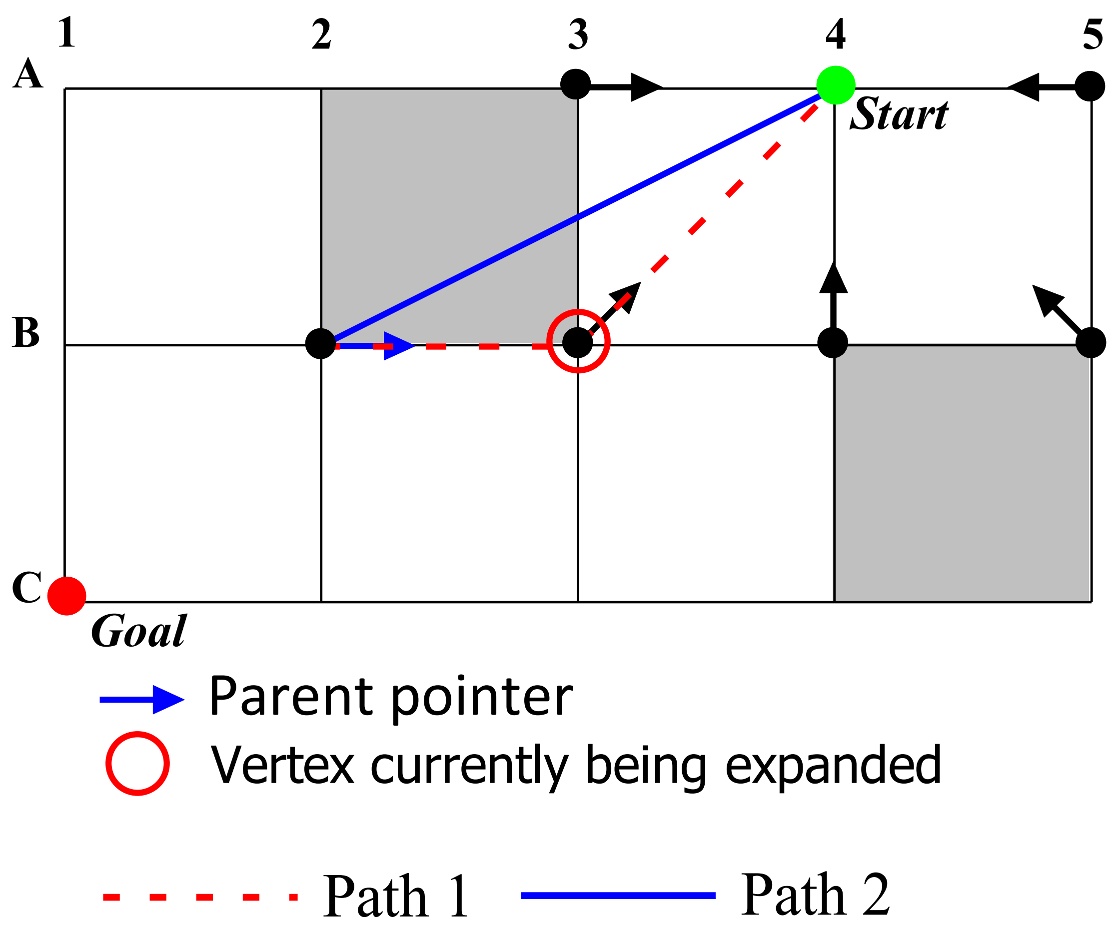
- Theta*



8-neighbor grid

Suboptimal Theta*

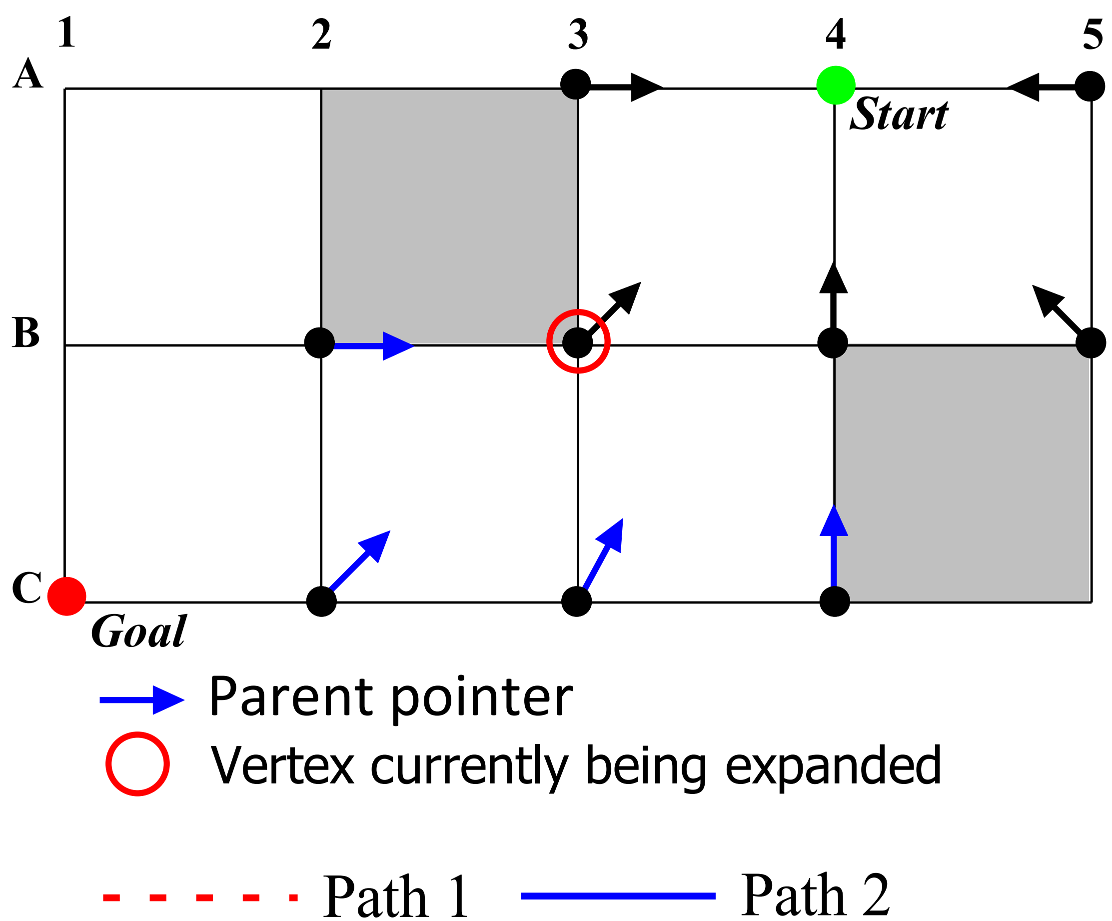
- Theta*



8-neighbor grid

Suboptimal Theta*

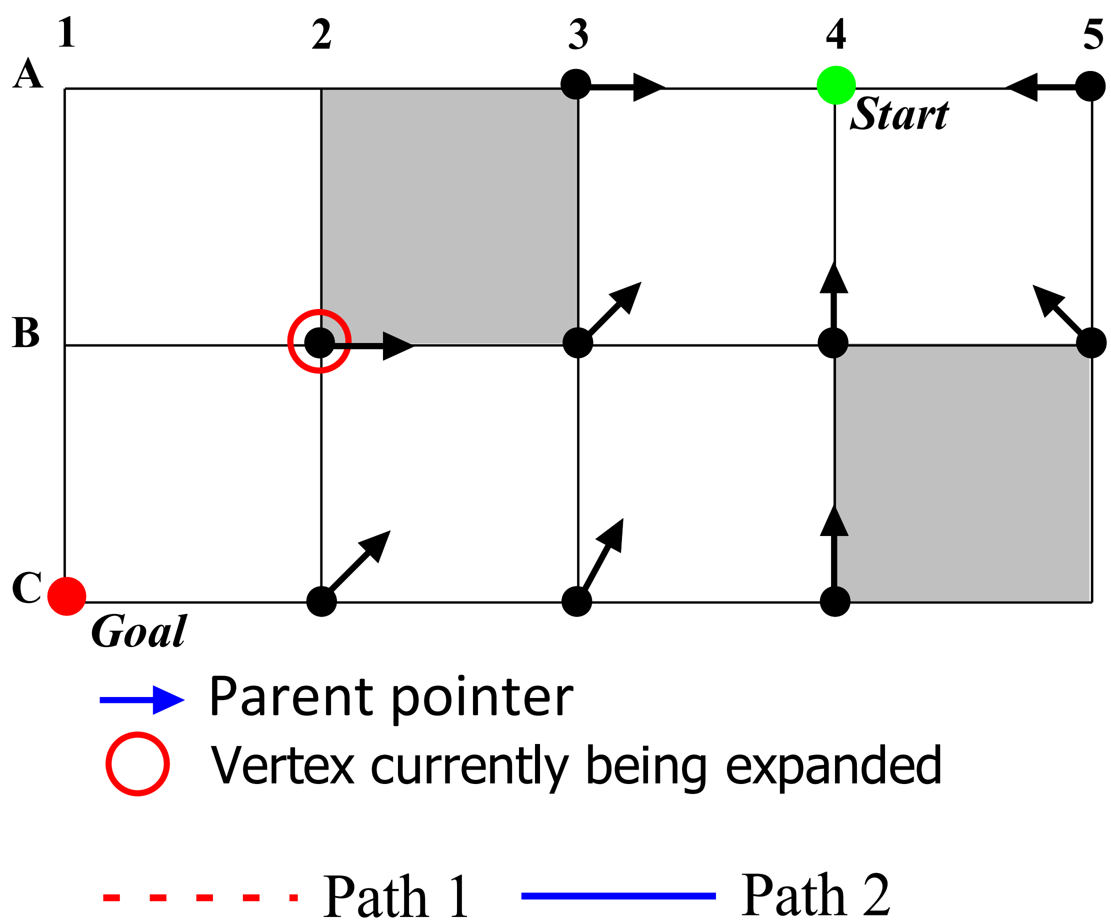
- Theta*



8-neighbor grid

Suboptimal Theta*

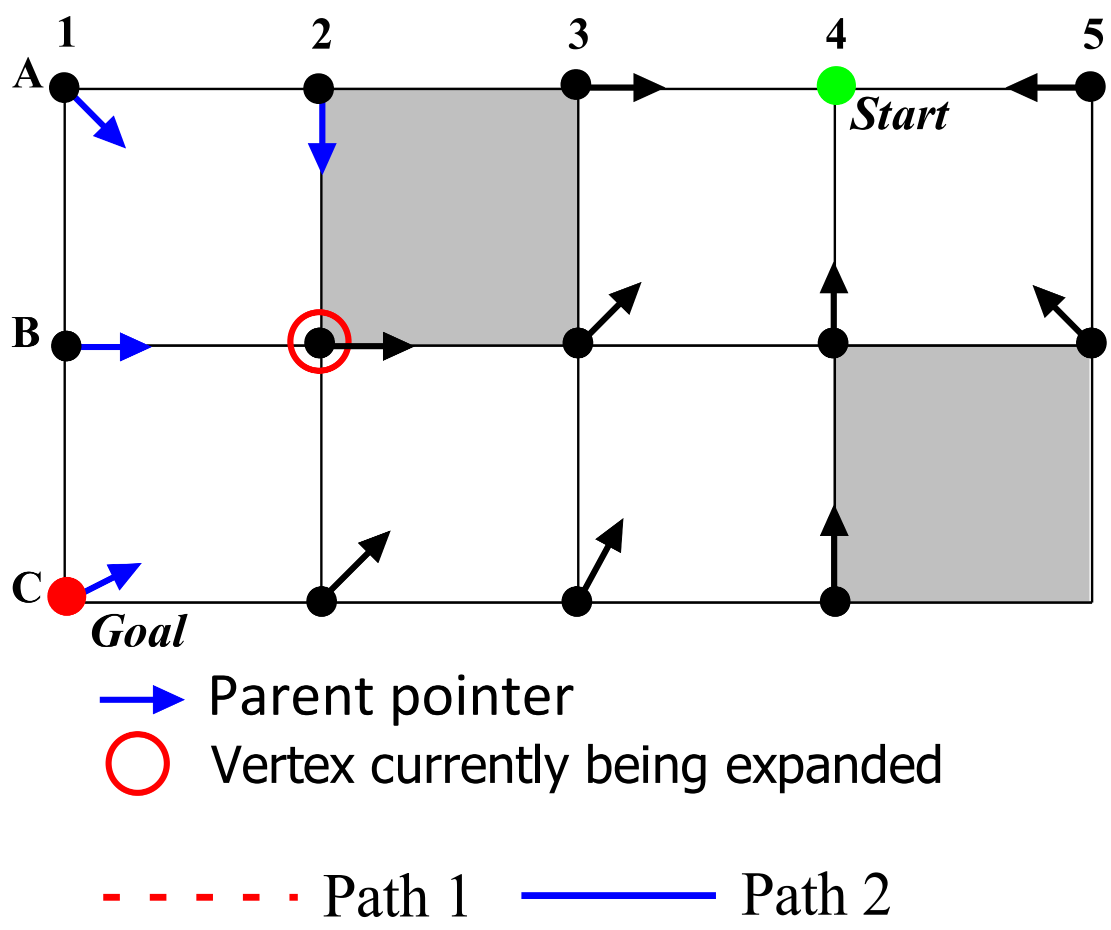
- Theta*



8-neighbor grid

Suboptimal Theta*

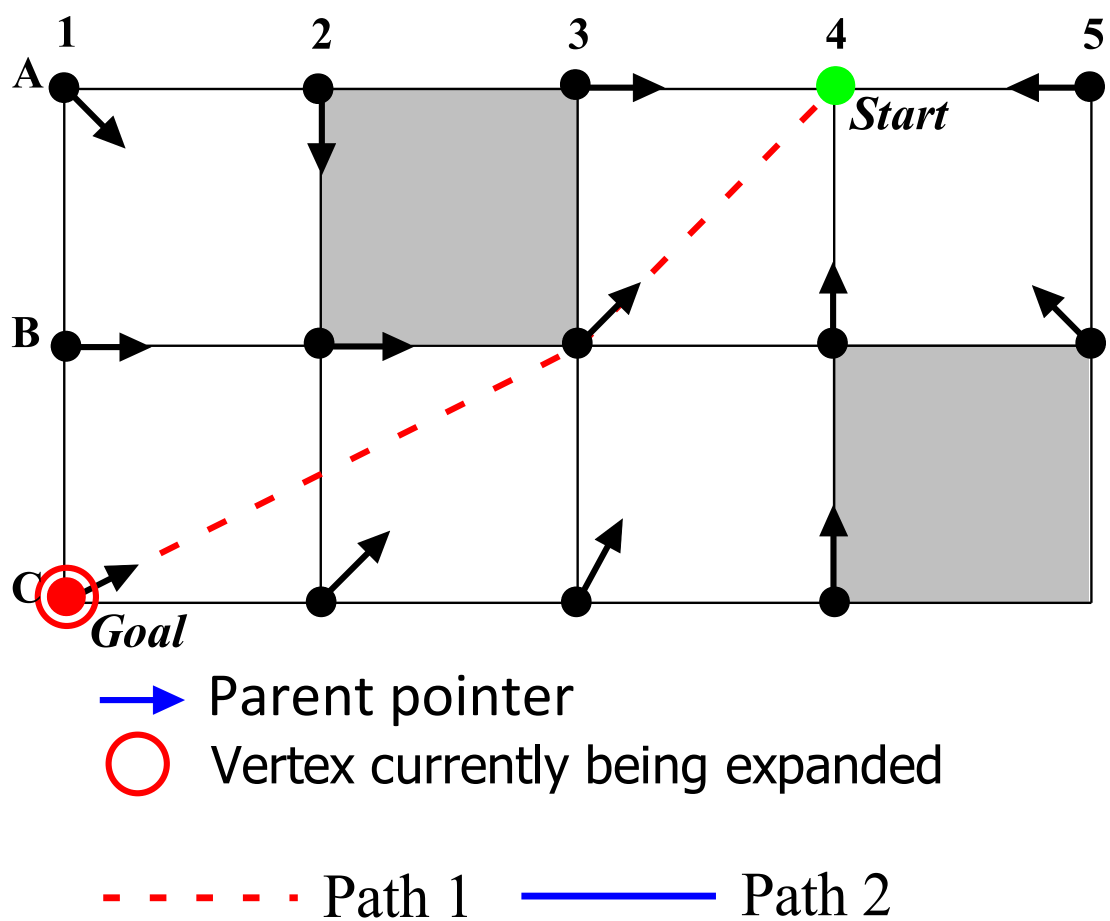
- Theta*



8-neighbor grid

Suboptimal Theta*

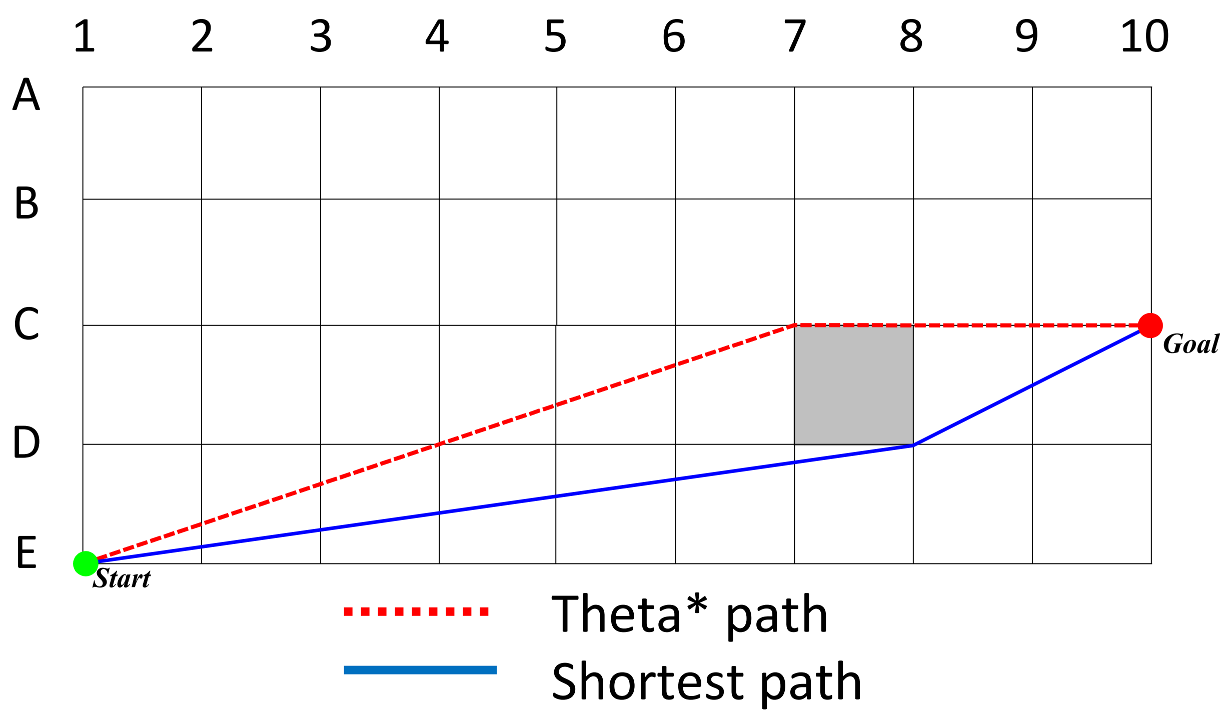
- Theta*



8-neighbor grid

Suboptimal Theta*

- Theta* is not guaranteed to find shortest paths since the parent of a vertex can only be a neighbor of the vertex or the parent of a neighbor



- The length of the path is still within 0.2% of optimal

8-neighbor grid

Suboptimal Lazy Theta*

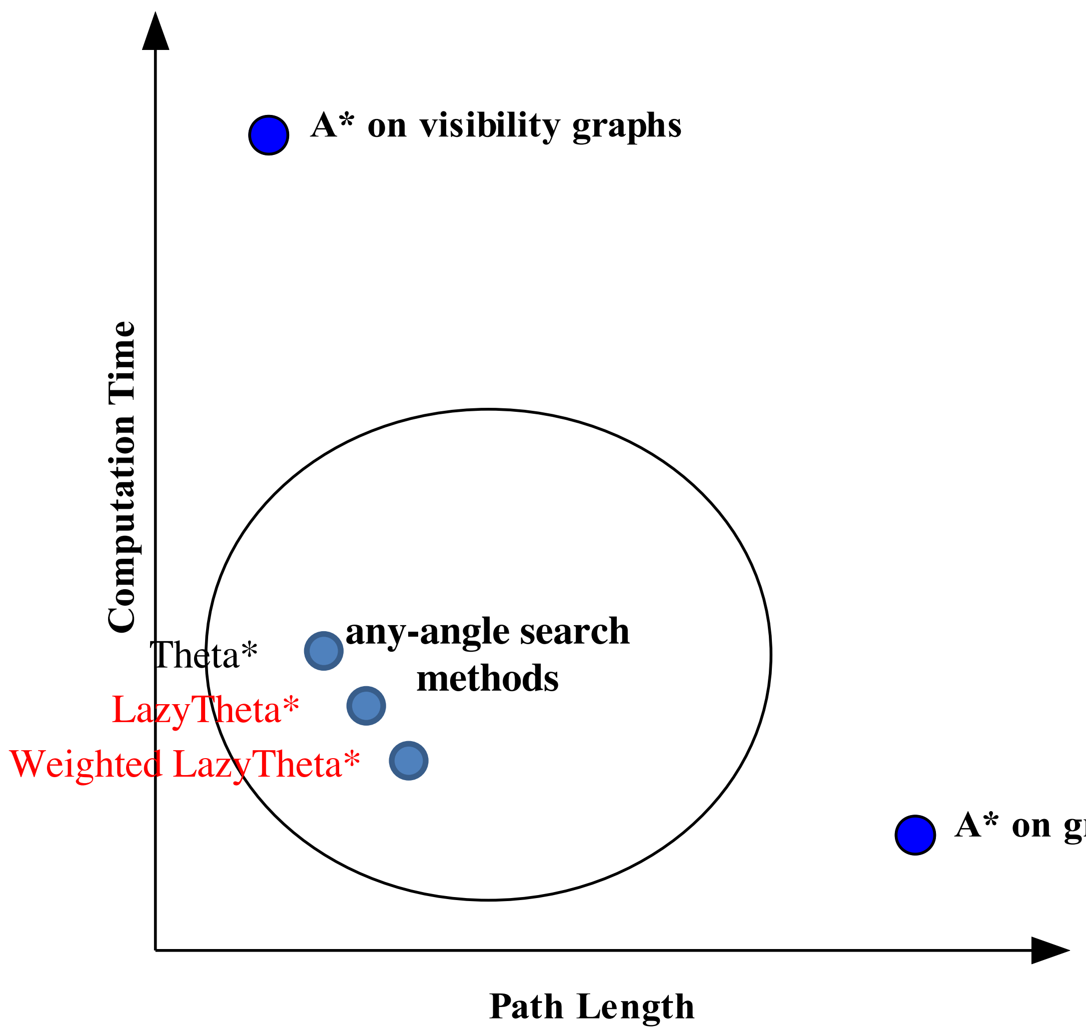
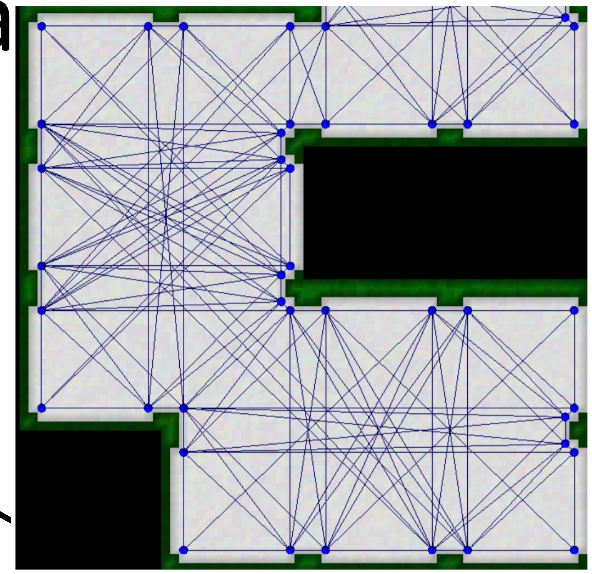
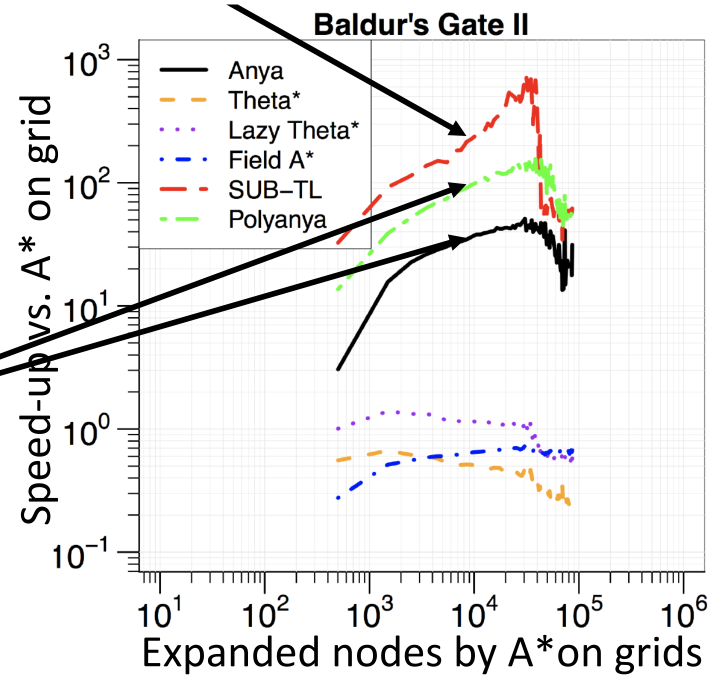


figure is notional

Alternatives to Theta*

- Other any-angle search algorithms
  - Several versions of Theta*: Lazy Theta*, Theta* on Subgoal Graphs (SUB-TL)
  - Accelerated A* [Sislak et al.] a sophisticated version of Theta*
  - Field D* [Ferguson and Stentz] an any-angle version of D* (Lite) with interpolation
  - Block A* [Yap et al.] an any-angle version of A* that operates on blocks of cells
  - Anya and Polyanya [Harabor et al.] any-angle search methods for 2D that find shortest paths





[from Zhao, Tania and Harabor]

[work done by different research groups]