Readings

- Path Planning: 255-275
- HPA*, SMA*, IDA*
Problems with Path Planning

http://www.ai-blog.net/archives/000152.html
Problems with Path Planning

http://www.ai-blog.net/archives/000152.html
Problems with Path Planning

http://www.ai-blog.net/archives/000152.html
Problems with Path Planning

http://www.ai-blog.net/archives/000152.html
Simplified Memory-Bounded A*

Just like A* but the open list has a max length.

Good for keeping a upper bound on memory consumption.

What's the problem with this approach?
Simplified Memory-Bounded A*

Just like A* but the open list has a max length.

Good for keeping a upper bound on memory consumption.

What's the problem with this approach?

Searches some areas more than once.
Iterative Deepening A*

Searches to an initial depth. If path not found, search to a larger depth.

"Does a recursive left-to-right depth-first search from the root node, stopping the recursion once the goal has been found or the nodes have reached a maximum value"

No open list means very small memory footprint.

Parameter is depth so it can be ran over different time slices.

What's the problem with this approach?
Fringe

An improvement on IDA*

Also searches to an initial depth. If path not found, search to a larger depth.

**Stores the frontier of the search!**
   Two lists store the fringe or frontier of search.

Good tradeoff of memory and speed to shortest path.

Very efficient in certain domains as it finds paths from two perspectives.
Theta*

Allows for paths of any angle from node to node.

What's new: any vertex can be the parent of any other vertex.

Uses line-of-sight (LoS)/visibility checks.
Theta*

http://aigamedev.com/open/tutorial/theta-star-any-angle-paths/
Theta*

Grid Path that is difficult to smooth

Grid Path that is easy to smooth

Shortest Path

http://aigamedev.com/open/tutorial/theta-star-any-angle-paths/
http://aigamedev.com/open/tutorial/theta-star-any-angle-paths/
“String pulling” style of aesthetic path enhancements are built in to the algorithm.

LoS checks are done on integers on the grid. Speedy!

Not guaranteed to find the shortest path!
Why: http://idm-lab.org/project-o.html
HPA* (Hierarchical A*)

Path plan on different levels of detail. Mipmapping for path planning.

Complete path at highest levels.

Only fill in lower levels of detail as needed or as processing power permits.
Cluster heuristic

- Can be good for indoor levels
- Pre-compute accurate heuristic between clusters
- Use pre-computed values for points in two clusters, Euclidean (or something else simple) within clusters
Examples on How to Build Hierarchy

http://aigamedev.com/open/review/near-optimal-hierarchical-pathfinding/
Excluding Nodes From Some Levels

Figure 4.40 Switching off nodes as the hierarchy is descended
Problematic Artifacts

Figure 4.41 Pathological example of the minimum method