**Algorithm for Exploration and Shortest Path Finding**

To qualify for Leaderboard A (Exploration)

Robot needs to go from Start Zone 🡪 End Zone 🡪 Start Zone WITHIN 6 mins

To qualify for Leaderboard B (Fastest Path), there are mainly 2 ways.

1. If you qualify for Leaderboard A, you automatically qualify for Leaderboard B.
2. If you don’t qualify for Leaderboard A, but robot manages to come *back* to Start Zone within 6 mins, you can challenge Leaderboard B.

(Stated under FAQ of Challenge rules)

As we discussed previously, shortest and fastest path does not necessarily mean the same thing.

For fastest path, it is said that we should “**pick expected travel time as weight for the edges instead of distance”. But I don’t think that this can this be applied in our scenario?**

**Source:** <http://stackoverflow.com/questions/3825142/fastest-path-algorithm>

**Maybe shortest path (Dijkstra’s Algorithm, “**searching the entire graph to compute its lowest-cost path. *For small graphs, this is feasible*, but for large graphs, the computing time just takes too long.”**) is sufficient since we *should* have the layout of the arena after exploration.**

**Source:** <http://research.microsoft.com/en-us/news/features/shortestpath-070709.aspx>

“They found that one of the most promising approaches was to use different techniques in combination, and then refine them.”

So far, it appears that A\* combined with Reach algorithm makes the best combination.

*“The so-called reaching algorithm can solve the shortest path problem (i.e., the problem of finding the graph geodesic between two given nodes) on an m-edge graph in O(m) steps for an acyclic digraph. This algorithm allows paths such that edges traversed in the direction opposite their orientation have a negative length.*

*No other algorithm can have better complexity because any other algorithm would have to at least examine every edge, which would itself take O(m)steps”*

**Source**: <http://www.siam.org/meetings/alenex04/abstacts/rgutman1.pdf>

<http://www.cs.princeton.edu/courses/archive/spr09/cos423/Lectures/reach-mit.pdf>

Another option is negative weight edges. Then the following algorithms can be considered.

**Shortest Path Faster Algorithm**

**Source**: <http://en.wikipedia.org/wiki/Shortest_Path_Faster_Algorithm>

**Others**

**Source**: <http://www.informit.com/articles/article.aspx?p=169575&seqNum=8>

**Exploration Algorithm**

Use either A\* or Maze Solving algorithm (Mainly pledge algo)

**Source**: [http://en.wikipedia.org/wiki/Maze\_solving\_algorithm#Pledge\_algorithm](http://en.wikipedia.org/wiki/Maze_solving_algorithm%23Pledge_algorithm)

<http://stackoverflow.com/questions/5361791/robot-exploration-algorithm>