Automatic Path Planning for Agricultural Irrigation Vehicles

Yunfeng Xin, Zhengyu Chen, Feng Chen

CS 221 Final Project Poster

Stanford Computer Science

Introduction

Coverage Path Planning(CPP) is a well-studied topic in recent years. Different approaches are applied to solve this type of problem using A* search, dynamic programming, and Q-learning to optimize CPP.

However, these solutions can lead to a number of unnecessary turns and can be costly for agricultural vehicles. Thus, We plan to evaluate and optimize some of these algorithms to avoid unnecessary turns as much as possible.

Model

Map: simplified as discrete grid blocks:
- Contain arbitrary number of obstacles modelling rocks and ponds

Agent: picks action in {up, down, left, right}

Goal: find a route such that the agent can traverse all the land to be irrigated:
- Covers all blocks
- Shorter distance
- Fewer turns

Challenges

Large search space:
- Infinite number of paths -> infinite search space
- Each location corresponds to 4 actions: branching factor = 4

No silver bullet:
- No single optimal algorithm
- Trade-off between running time, #turns, and total distance

Approach

A* Search:
- Go straight if possible
- When agent is trapped with visited blocks, use A* to generate a path to an unvisited block

Depth-limited Search:
- Try all the paths within limited depth
- Choose the action with the highest possible rewards
- When trapped, use A* to generate a path to unvisited block

Local-approximation:
- Choose the action based on a smaller local map feature
- When trapped, use A* to generate a path to unvisited block

Gradient Descent:
- Need goal location
- Preprocess the map and label grid with its "height" (distance to goal)
- Always follow the highest grid
- Use auxiliary algorithm (A* or tracing back) to escape local traps

TD-learning:
- Use neural network to generalize features
- Stochastic gradient decent update
- ϵ-exploration
- Use A* to escape local traps

Result

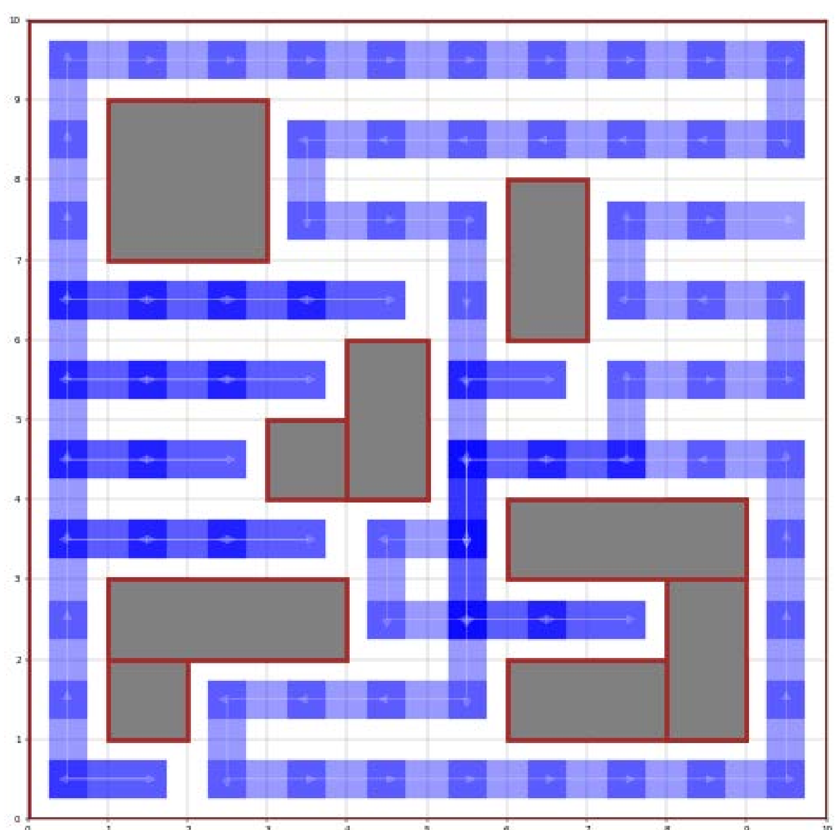

Fig 1. Depth-limited Search with A*

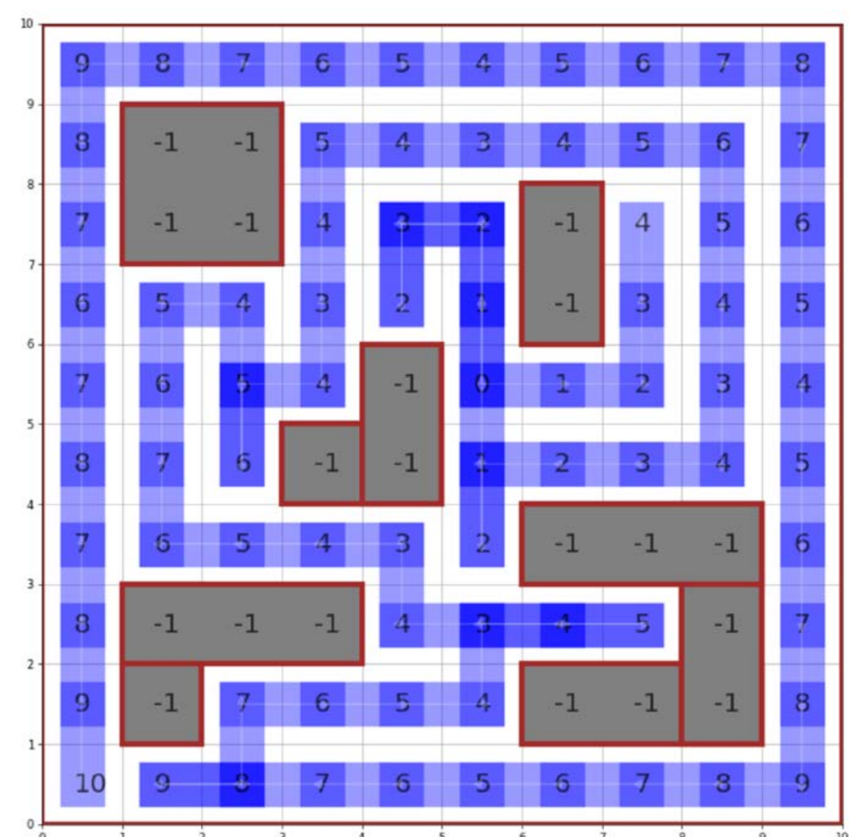

Fig 2. Gradient Decent without A*

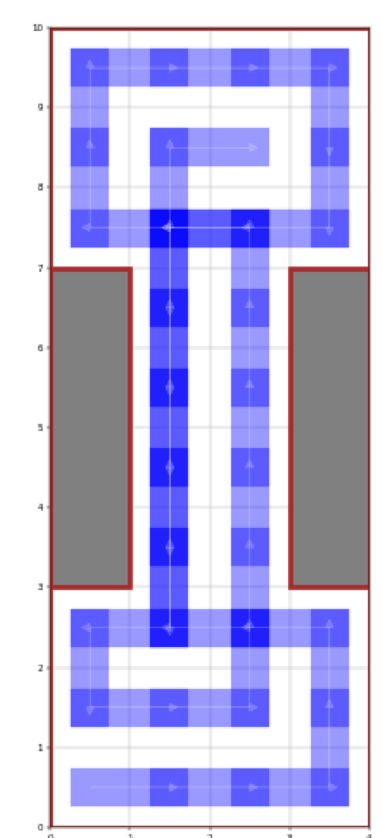

Fig 3. A* in H-Shaped Map

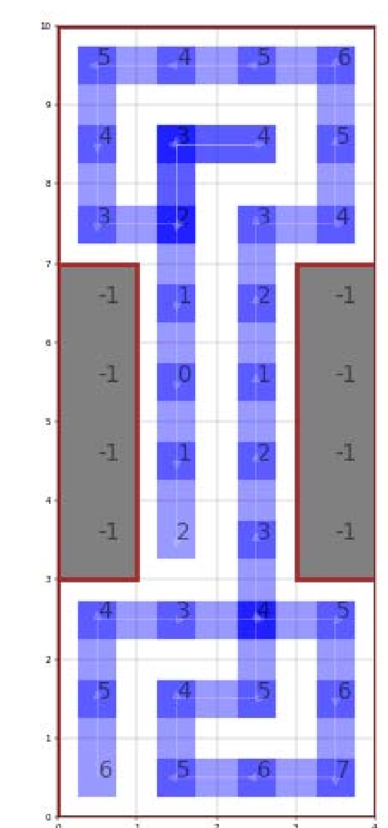

Fig 4. Gradient Decent with A*

Map	Algorithm	Turn number	Total Distance
Complex Map	Depth-limited Search with A*	41	103
	Local-approximation Search with A*	44	124
	A*	34	137
	Gradient Decent with A*	32	104
	Gradient Decent without A*	29	89
	TD-learning with A*	42	138
H-shaped Map	Depth-limited Search with A*	14	41
	Local-approximation Search with A*	17	39
	A*	13	64
	Gradient Decent with A*	15	35
	Gradient Decent without A*	17	36
	TD-learning with A*	15	38

Analysis

A* search Algorithm:
+ Guaranteed full traversal
+ Require less computation
– Unnecessary block revisit.
– Wasted distance and turns in the path from trapped cell to unvisited.

Depth-limited Search:
+ Asymptotic optimal solution
+ Require cost function to balance between turns and distance
– Extensive computation with large limited depth
– May be trapped without A* guidance

Local-approximation:
+ Asymptotic optimal solution
+ Require cost function to balance between turns and distance
– Extensive computation with large local map
– May be trapped without A* guidance

Gradient Descent Based Algorithm:
+ Guaranteed full traversal
+ Little online computation
+ Balanced between turns and distance
– Unstable results with different goals
– Require precomputation

TD-learning:
+ Balanced between computation and optimization
+ Cost function can balance between turns and distance
– Unstable algorithm; results may vary from time to time
– May be trapped without A* guidance

References

[1] A. Ntawumenyikizaba, Hoang Huu Viet and TaeChoong Chung, "An online complete coverage algorithm for cleaning robots based on boustrophedon motions and A* search," 2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012), Jeju, 2012, pp. 401-405.

[2] Galceran, Enric, and Marc Carreras. "A survey on coverage path planning for robotics." Robotics and Autonomous systems 61.12 (2013): 1258-1276