Artificial Intelligence

Lesson 12

Many applications:
- Floor cleaning, mowing, de-mining, ...

Many approaches:
- Off-line or On-line
- Heuristic or Complete

Multi-robot, motivated by robustness and efficiency

Robotics, a Case Study - Coverage

Environment Assumptions
- Static - to be able to guarantee completeness
- Inaccessible - greater impact on the on-line version
- Non-deterministic
- Continuous
  - Exact cellular decomposition
  - Approximate cellular decomposition

MSTC - Multi Robot Spanning Tree Coverage
- Complete - with approximate cellular decomposition
- Robust
  - Coverage completed as long as one robot is alive
  - The robustness mechanism is simple
- Off-line and On-line algorithms
  - Off-line:
    - Analysis according to initial positions
    - Efficiency improvements
  - On-line:
    - Implemented on simulation of real-robots
Off-line Coverage, Basic Assumptions

- Area division – \( n \) cells
- \( k \) homogenous robots
- Equal associated tool size
- Robots movement

STC: Spanning Tree Coverage

(Gabrieli and Rimon 2001)

- Area division
- Graph definition
- Building the spanning tree

Non-backtracking MSTC

- Initialization phase: Build STC, distribute to robots
- Distributed execution: Each robot follows its section
  - Low risk of collisions

Guaranteed Robustness

- Coverage completed as long as one robot is alive
- Low communication, no need for re-allocation
Analysis: Non-backtracking MSTC

- Running time = $\max_{i,k} \text{step}(i)$

- Best case: $\left\lceil \frac{n}{k} \right\rceil - 1$

- Worst case: $n - k$
  - Unfortunately, common case

- General non-backtracking worst case: $n - 2(k-1) - 1$

Backtracking MSTC

- Similar initialization phase
- Robots backtrack to assist others
- No point is covered more than twice

Backtracking MSTC (cont.)

- Same robustness mechanism
- Same communication requirements

Backtracking MSTC Analysis

Best case: The same

Worst case:
\[
\begin{cases}
  k=2 & \left\lceil \frac{2n}{3} \right\rceil \\
  k>2 & \left\lceil \frac{n}{2} \right\rceil
\end{cases}
\]
Efficiency in Off-line Coverage

- Optimal MSTC - improves the average case
- Heterogeneous robots - flexibility
- Optimal spanning tree - improves the worst case

Optimal MSTC

- Similar initialization phase
- Robots backtrack to assist others:
  - All the robots can backtrack
  - Backtracking on any number of steps
- No point is covered more than twice

- Same robustness mechanism
- Same communication requirements

Optimal MSTC (cont.)

- Choose a robot
- Search for the minimum valid solution
  - Left search
  - Right search
- Complexity:
  - Check on all the robots: k
  - Each search: O(n logn)
  - Validity check: O(k)
  - Total: O(k^2 logn)

Heterogeneous Robots

- Different speeds
  - Non-backtracking MSTC
  - Backtracking MSTC
  - Optimal MSTC

- Different fuel/battery time
  - Non-backtracking MSTC
  - Backtracking MSTC
  - Optimal MSTC
Optimal Spanning tree

- Improves the worst case with all 3 algorithms
- The construction is believed to be NP-Hard

A Heuristic Solution

- Build $k$ subtrees on coarse grid
  - Start building subtrees from initial locations
  - Add cells to each subtree gradually
  - Spread away from other robots (based on Manhattan dist)
- Connect subtrees
  - Randomly pick connections between subtrees
  - Calculate $x$ in resulting tree
  - Repeat $k^\alpha$ times ($\alpha$ is a parameter)
  - Report tree yielding minimal $x$

Illustration – Stage 1

Generating a Good Spanning Tree (Believed to be NP-Hard)
Example

On-line MSTC

- Same basic assumptions:
  - Area decomposition - $n$ cells
  - $k$ homogenous robots
  - Equal tool size and robot movements
- All the robots know their absolute initial position
- Initialization phase
  1. Agreed-upon grid construction
  2. Self-localization
  3. Locations update

On-line MSTC (Cont.)

Guaranteed Robustness

- Coverage completed as long as one robot is alive
- No need for re-allocation
From Theory to Practice

- Player/Stage with modeled RV-400 robots
- Localization solutions
  - GPS
  - Odometry with limited errors
- Agreed-upon grid options
  - Big enough work-area
  - Dynamic work-area
- Collisions avoidance with bumps
  - Random wait
  - Communication based
- Limited sensors solution

Off-line Algorithms Experiments (1)

- Work area: 30X20 cells, 2400 sub-cells
- Each point represents 100 trials

Off-line Algorithms Experiments (2)

- Work area: 30X20 cells with 80 holes, 2080 sub-cells
- Each point represents 100 trials

Experimental Results
Experimental Results - 27% Obstacles

On-line Algorithm Run-time Example

On-line Algorithm Experiments

- Random places
- Each point represents 10 trials

Conclusion

- Complete and robust multi-robot algorithms
- Redundancy vs. efficiency with off-line algorithms
- Optimal MSTC which handle heterogeneous robots
- Implemented on-line MSTC with approximation techniques