# Distributed Path Planning for Mobile Robots using a Swarm of Interacting Reinforcement Learners

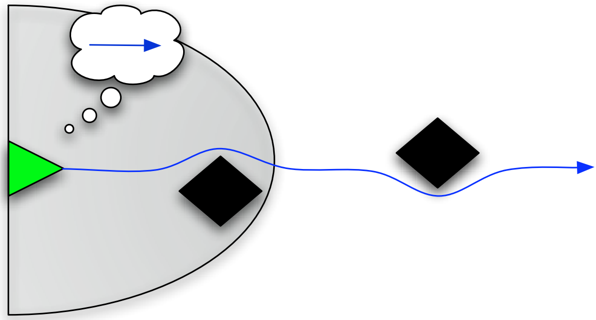
Chris Vigorito

Department of Computer Science
University of Massachusetts - Amherst
vigorito@cs.umass.edu

May 17th, 2007
AAMAS '07 - Honolulu, HI

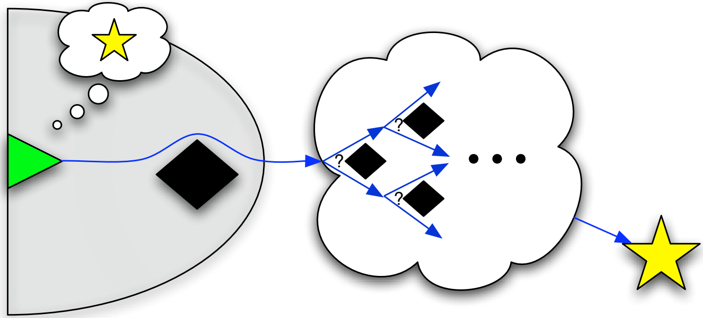
# Local Robot Navigation - Obstacle Avoidance

- Local navigation (obstacle avoidance)
  - Goal observable or only heading given
  - Head in desired direction while avoiding obstacles
  - Reasonably good approaches for solving this problem



# Global Robot Navigation - Path Planning

- Global navigation (path planning)
  - Goal unobservable/heading unknown (need a model)
  - Want least cost path to goal
  - Lots of uncertainty/decision points
  - Some egocentric approaches with restrictive assumptions and high complexity



# Physical Path Planning

- Places computational burden on distributed sensor network rather than on robot
- Network of unsophisticated sensor nodes with local communication capabilities
- Nodes communicate path information locally to produce globally optimal solution
- Low complexity computation at each node
- Robots query nodes for least cost path to desired goal

# Previous Work

- Perform distance vector routing over topological map formed by sensor network
- Cost-metrics used limited to hop count [Batalin, et al. (2004); Li et al. (2003); O'Hara, et al. (2006)]
- Nodes must be able to sense relevant information
- No information from robot experience is used (no learning)
- Only tested on uniform terrain with highly structured (e.g., grid-like) network deployments
- Contribution: Incorporate reinforcement learning to improve solution quality and versatility

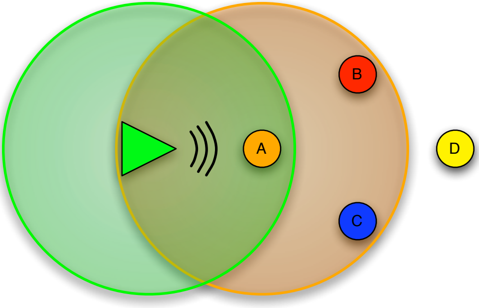
# Distance Vector Routing

- Route incoming packet/robot to next hop router so as to minimize cost function given a destination
- Each node stores a distance vector estimate (estimated cost from self to all destinations)
- $$D(x,z) = \min_{y \in N(x)} d(x,y) + D(y,z)$$
- Distributed form of Bellman-Ford algorithm (dynamic programming)
- Widely used in networking applications

# Reinforcement Learning

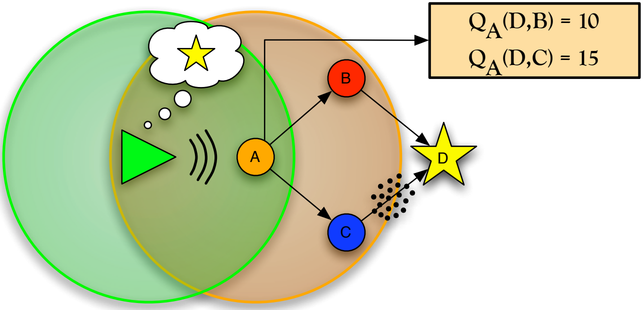
- Well developed framework for learning from experience how to interact with an environment
- Goal is to maximize reward (here to minimize a cost function)
- Common formalism: Markov Decision Process (MDP) = $< S, A, T, R >$
- Agents learn policy $\pi$ to map states to actions that minimize cost function
- Can be solved by learning an action-value function $Q : S \times A \rightarrow \Re$
- Network routing problem formulated as MDP in Boyan and Littman (1993)

# Model and Assumptions



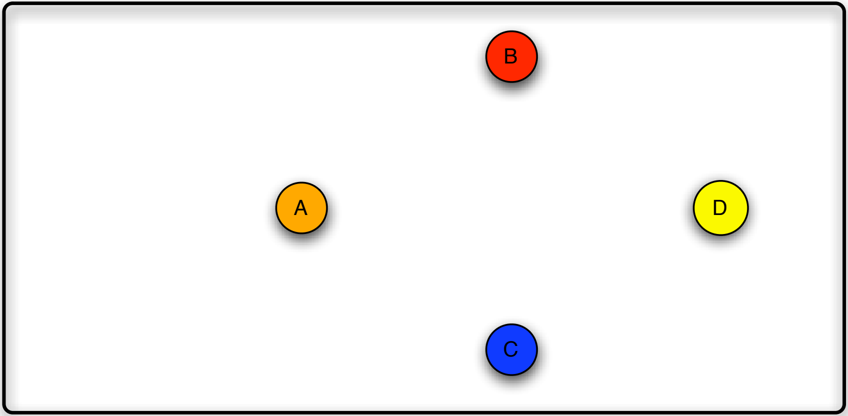

- All nodes/robots have some means of local communication
- All robots equipped with local navigation abilities
- All robots can obtain distance and heading to a nearby node

# Swarm of Interacting Reinforcement Learners (SWIRL)

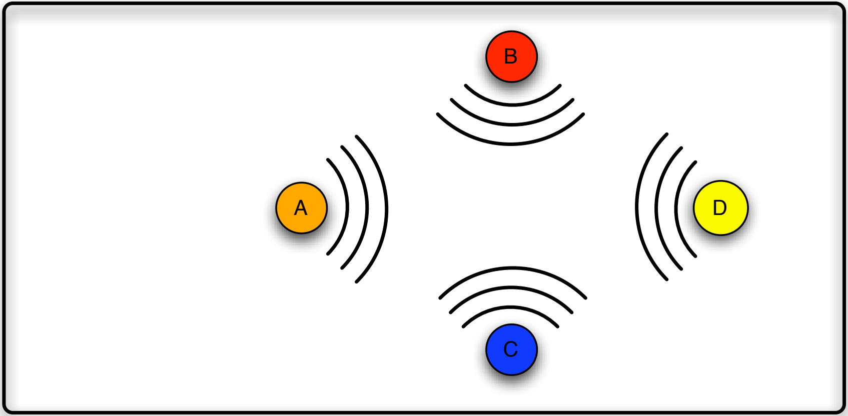



- States represented as node/destination pairs
- Actions are next hop choices
- Transition function defined by network topology
- “Reward" = time, energy, danger, etc.
- Value function distributed across network

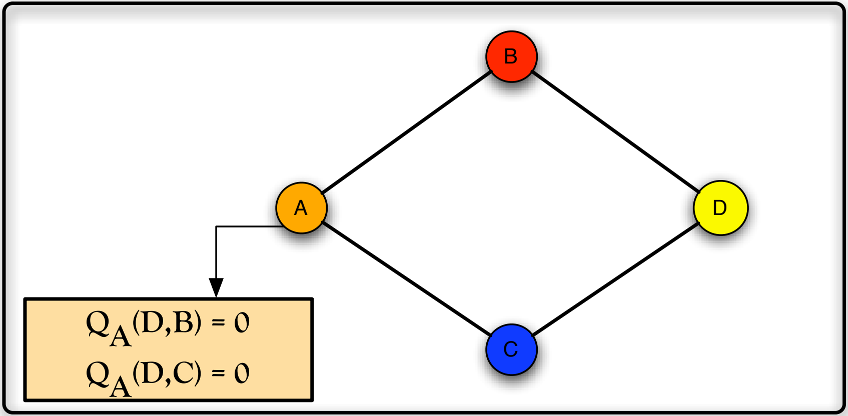
# Algorithm



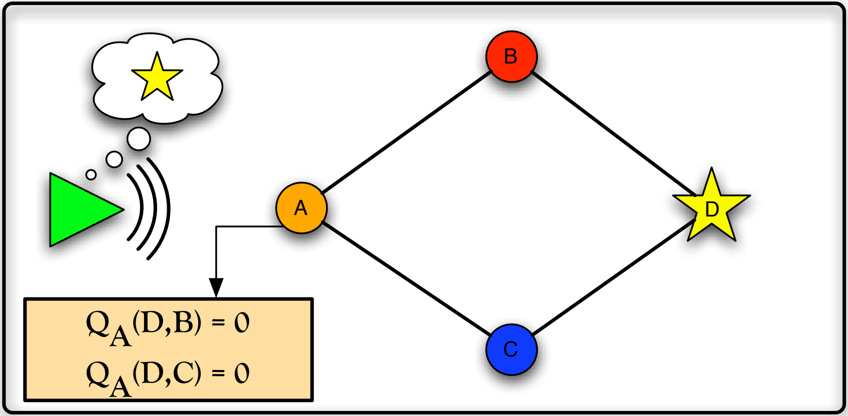
# Algorithm



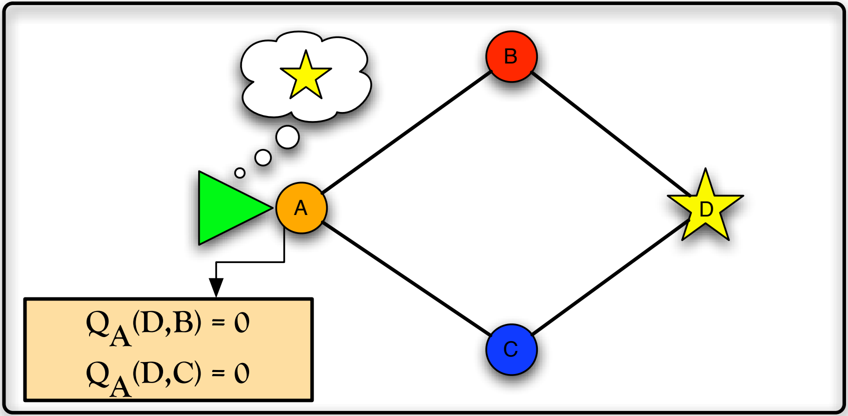
# Algorithm



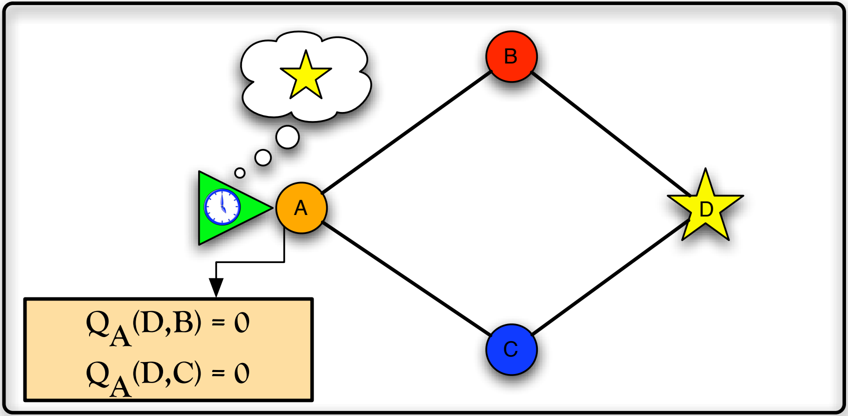
# Algorithm



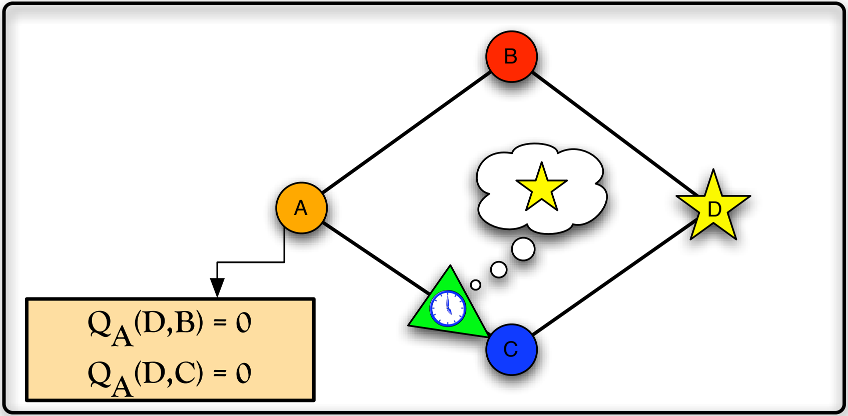
# Algorithm



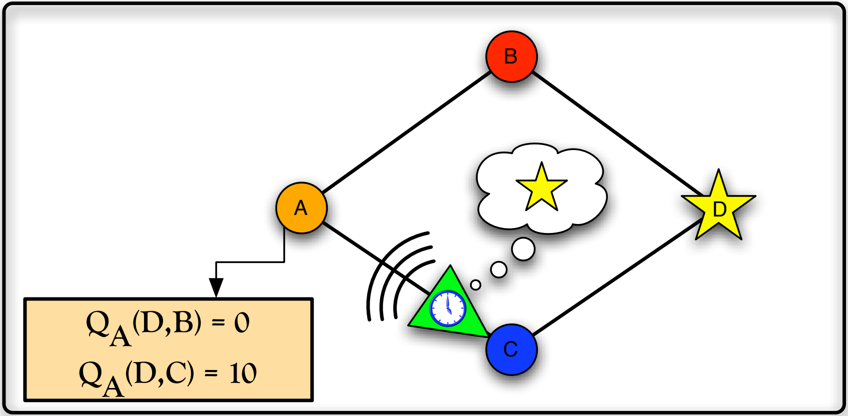
# Algorithm



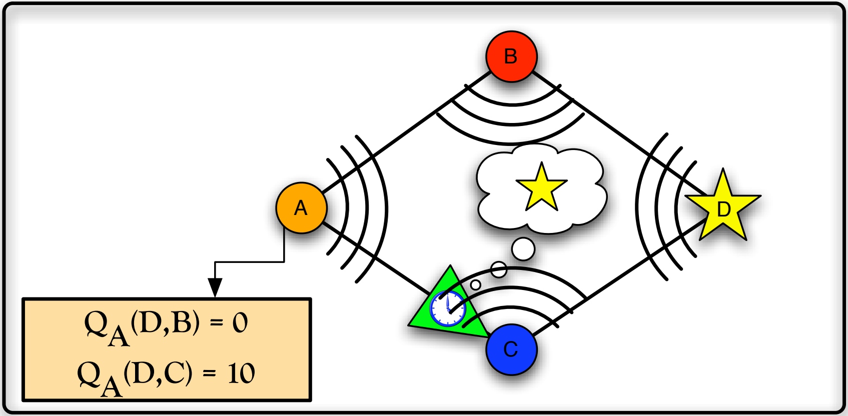
# Algorithm



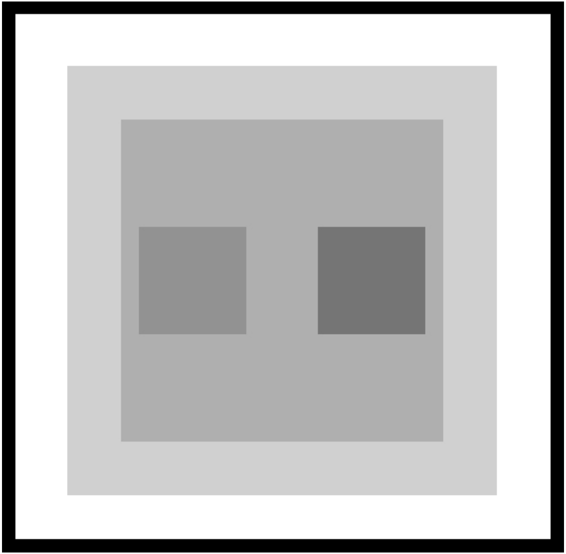
# Algorithm



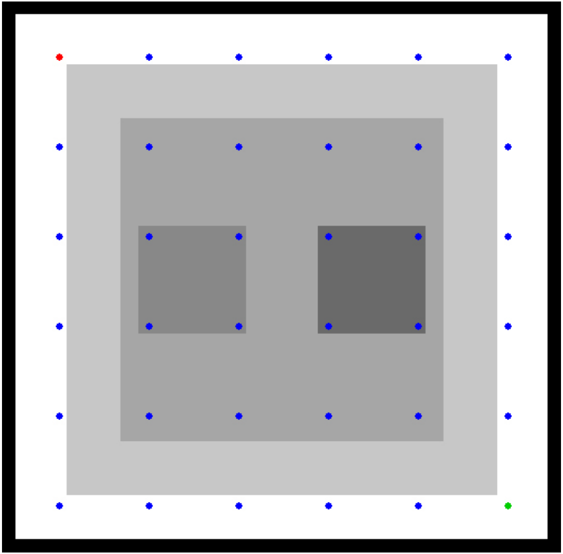
# Algorithm



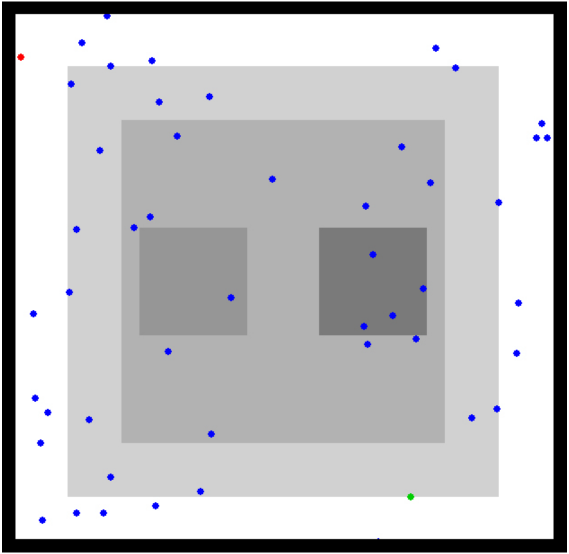
# Simulation Environment



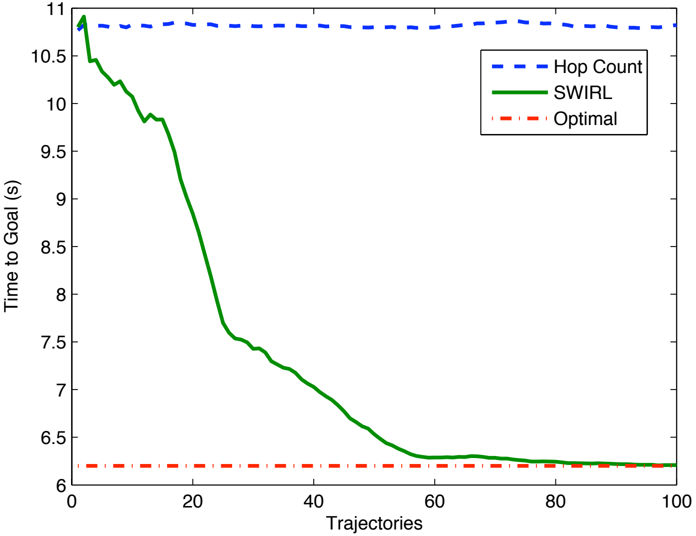
# Grid Network Deployment



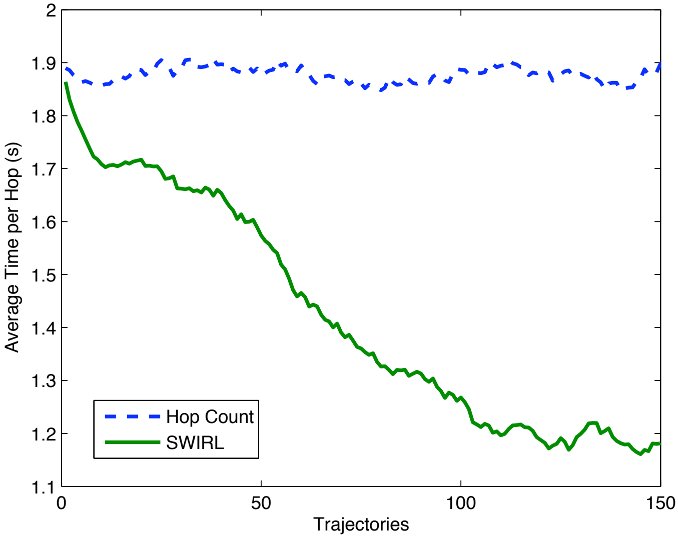
# Random Network Deployment



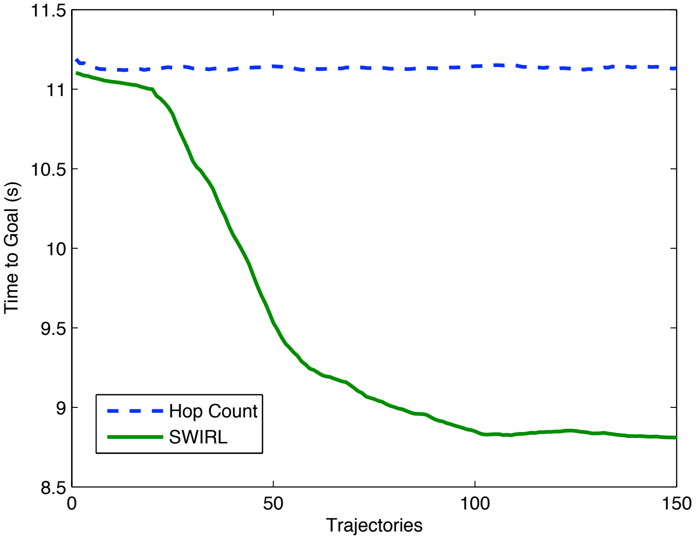
# Grid Deployment - Single Start/Goal Pair



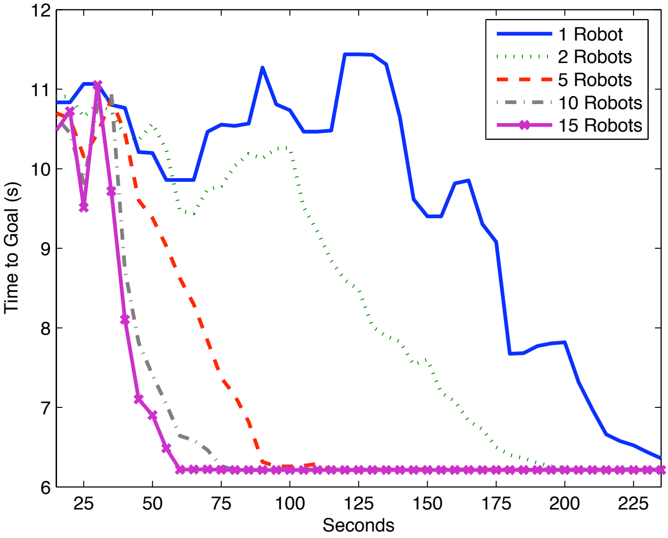
# Grid Deployment - Random Start/Goal Pairs



# Random Deployment - Single Start/Goal Pair



# Grid Deployment - Single Start/Goal Pair



# Summary

- Extension of existing methods for physical path planning
- Incorporated reinforcement learning to improve solution quality in the face of unobservability/uncertainty
- Performs well in wider class of environments
- Allows for less structured types of network deployments

# Limitations and Future Work

- Approach doesn't currently address situation in which links are not traversable
- Add ability for robots to sense impasses and send infinite edge weights to nodes
- Mobility of sensor nodes - reconfiguration for better coverage
- Have robots use "shortcuts" by interpolating between nodes