# CMRoboBits:
# *Probabilistic Path Planning*

Manuela Veloso

15-491, Fall 2008

http://www.andrew.cmu.edu/course/15-491

Computer Science Department

**Carnegie Mellon**

# Problem Solving - Planning

- Allen Newell and Herb Simon – 1950s:
  - Problem solving/planning:
    - Given an initial state, a set of action, a goal statement
    - Find a sequence of actions that transform the initial state into a state where the goal is satisfied
- *Path* planning:
  - Continuous state space
  - Motion actions

# General Search

- General Search (*problem*, *strategy*)
  - Initialize state, goal, actions from *problem*.
  - If there are candidate states
    - Choose a state according to *strategy*
    - If the goal is in the state,
      - return success and solution
      - otherwise, expand the state, i.e., generate successor to state as new candidate states
  - Otherwise return failure

# Strategy

- Depth-first search
- Breadth-first search
- A* (read paper)
- Probabilistic path planning

# Path Planning

- Existence of a goal
  - *Goto* some goal point
- ERRT
  - Efficient Rapidly-Exploring Random Tree
  - Path planning
  - Smoothing
  - Memory

# Motion Planning

- Motion planning
  - Finding a path from a source to a target
  - Subject to constraints
    - From the environment (obstacles)
    - From the robot's capabilities
- Requirements for motion planning
  - An environmental model
  - An action model

# Environment Models

- An environment model is composed of
  - Knowledge of the robots location (Localization)
  - Knowledge of the existence of location of obstacles

- Complicating factors
  - Number of dimensions
  - Number of obstacles and complexity of geometry
  - Complexity of robot state
  - Error or uncertainty from sensors

# Action Models

- Action models
  - Knowledge of how an action affects the environment
  - Must be known *without* executing the action

- Complicating factors
  - Constraints on robot actions
    - Motion (kinematic) constraints (e.g. car-like robots)
    - Bounded velocity and acceleration
  - Dynamics effects at high speeds
  - Error or uncertainty in actions

# Distance Scales for Planning

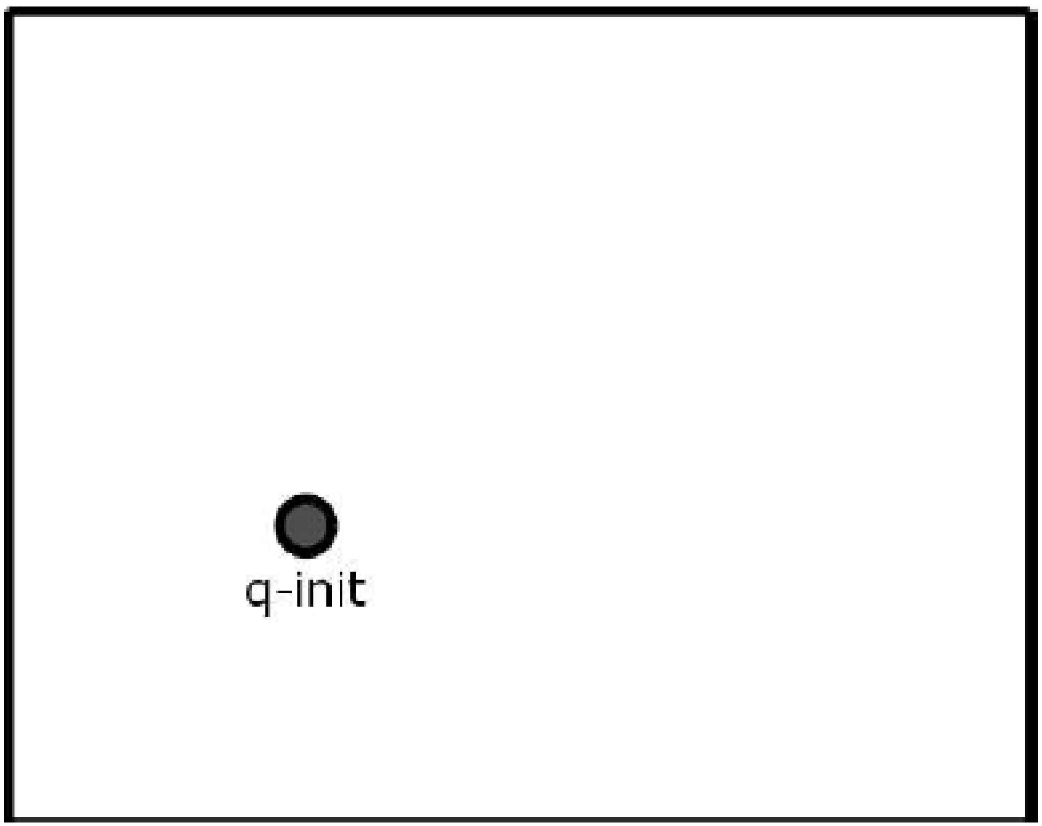
- Distance scales for planning
  - Local
    - Kinematic and dynamic constraints must be respected
  - Long range
    - Essentially just path planning
  - Intermediate
    - Some features of both local and long range
  - Indoor robots are mostly local to intermediate

# One Motion Planning Approach: RRT

- Rapidly Exploring Random Trees (RRT)
  - Explore continuous spaces efficiently
    - No need for an artificial grid
  - Form the basis for probabilistically complete planners
    - Some change of finding a solution if it exists
- Complete planners exist, but are far too slow
- RRT uses random search and approximation for speed

# Basic RRT Example

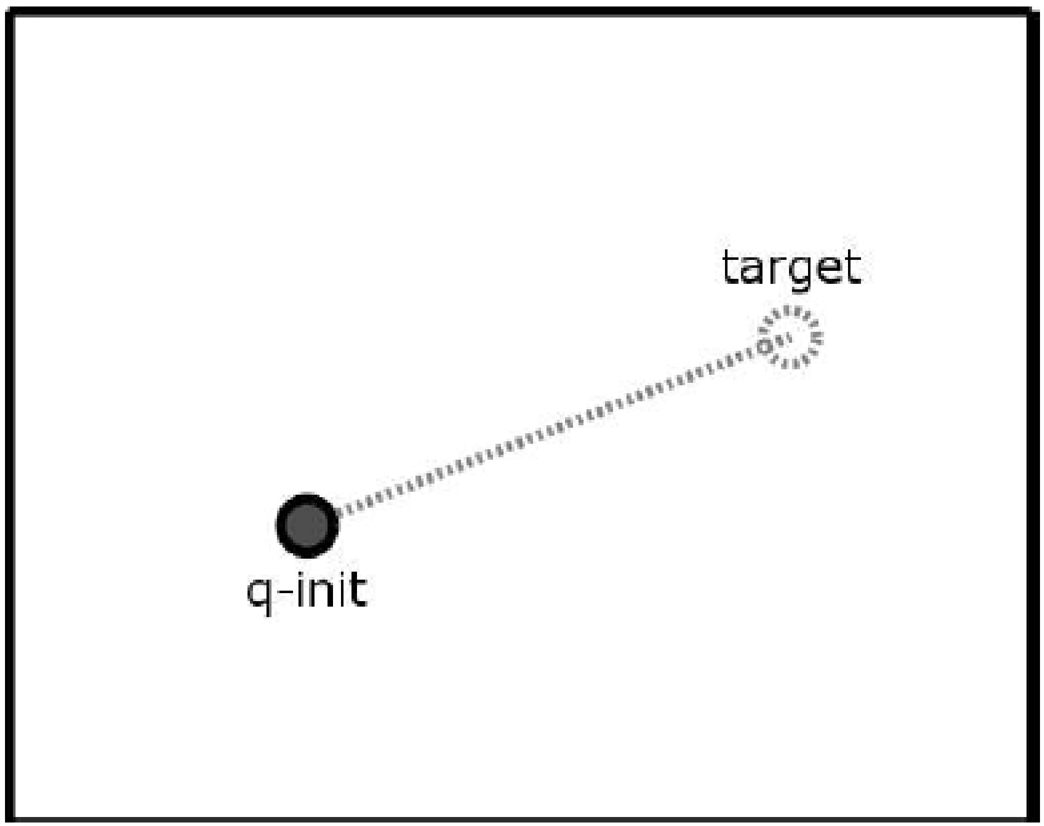
(1) Start with the initial state as the root of a tree



# Basic RRT – Just Search, No Goal

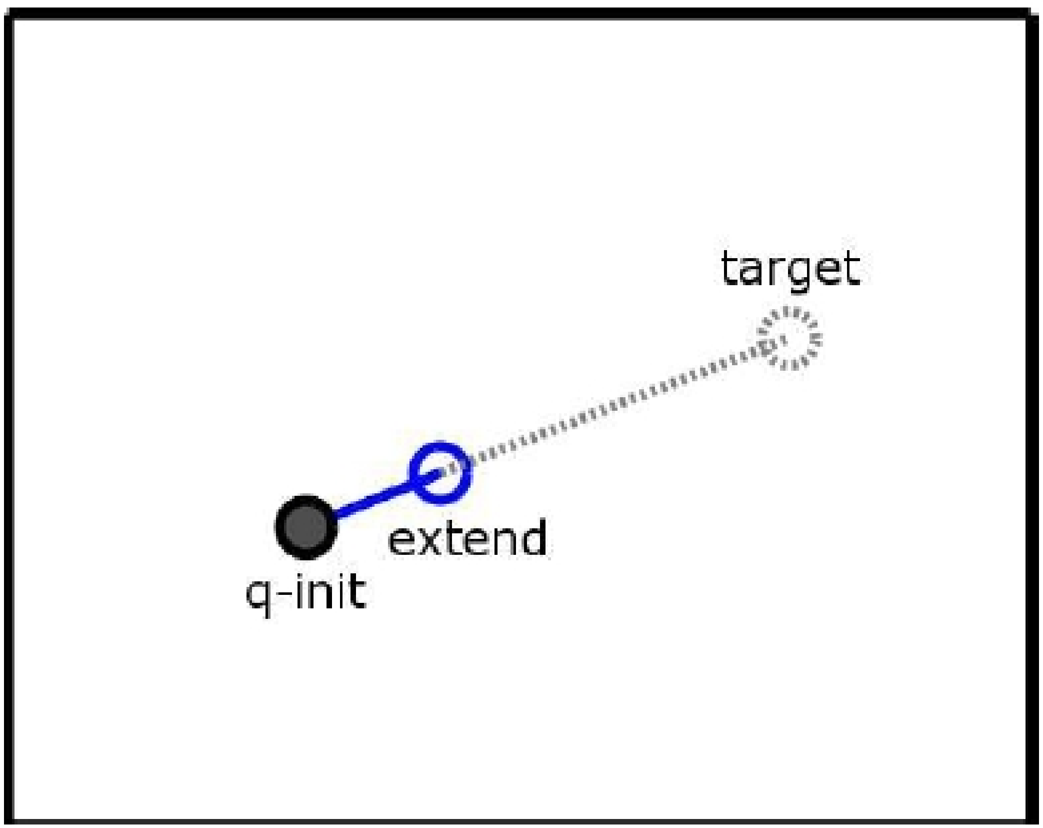
(2) Pick a random state in the environment

(3) Find the closest node in the tree

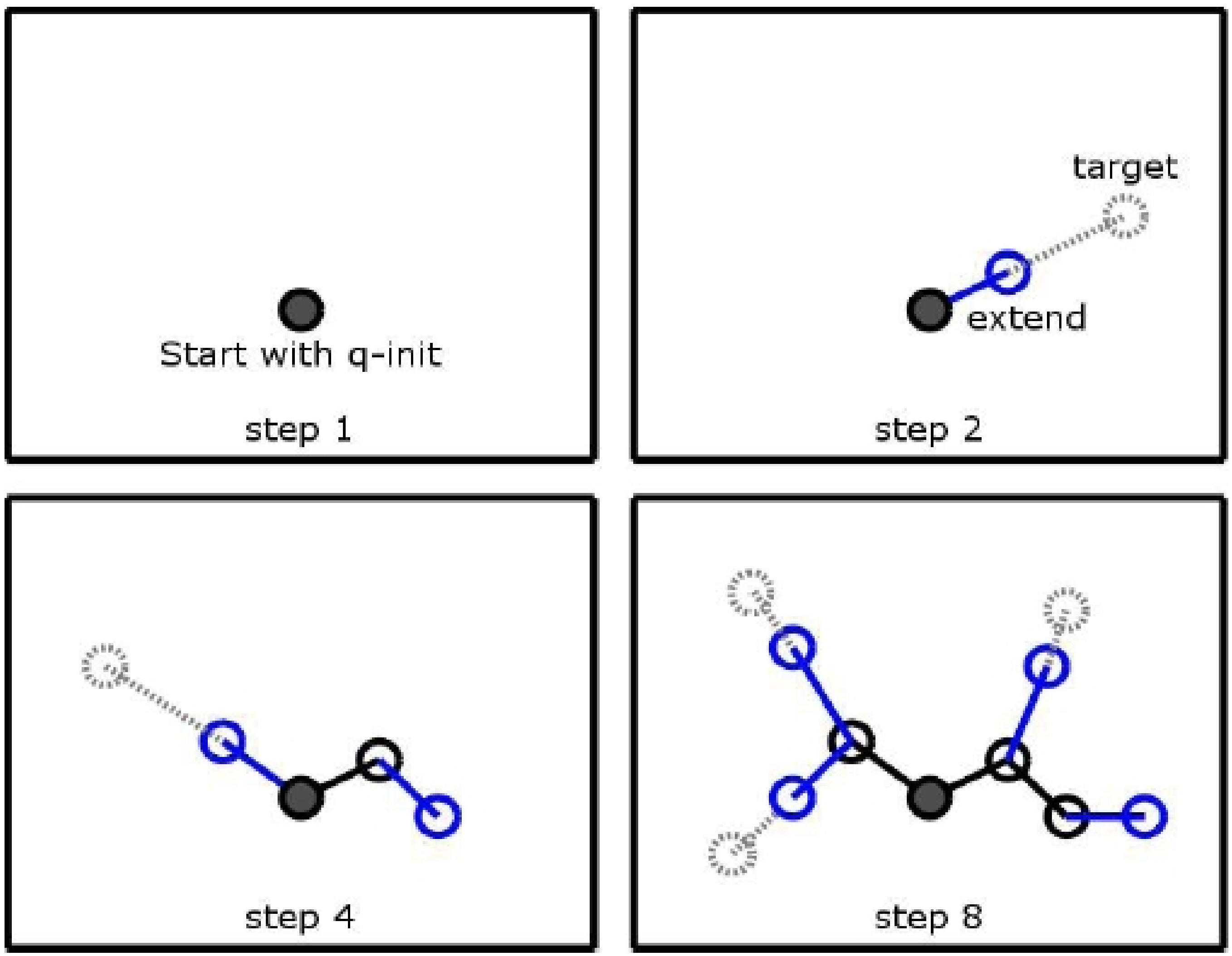


# Basic RRT Search – No Goal (cont.)

(4) Extend that node toward the target if possible

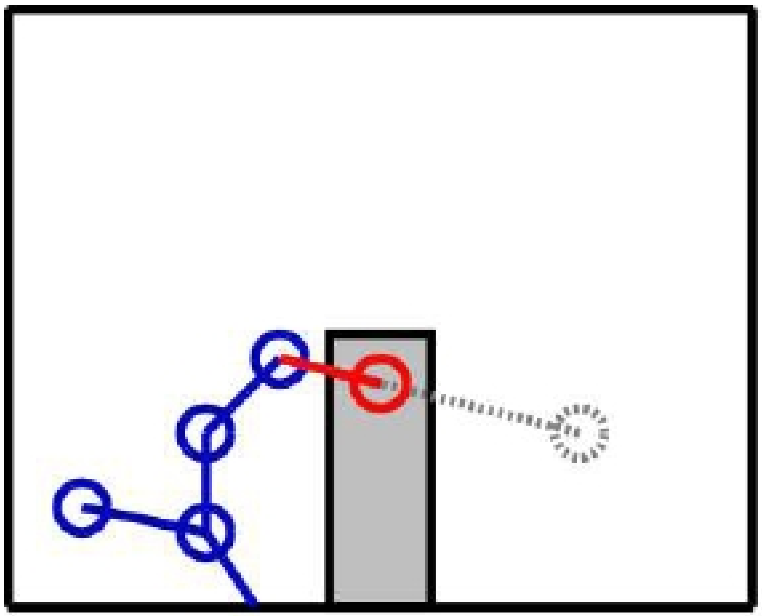


# Basic RRT Search (no Goal) Summary

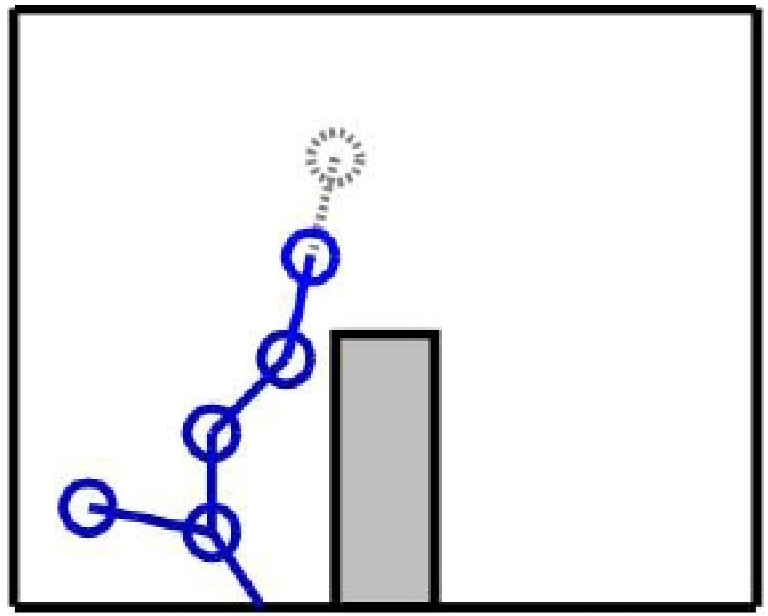


# RRT with Obstacles

- Ignore extensions which hit obstacles
- Resulting tree contains *only* valid paths



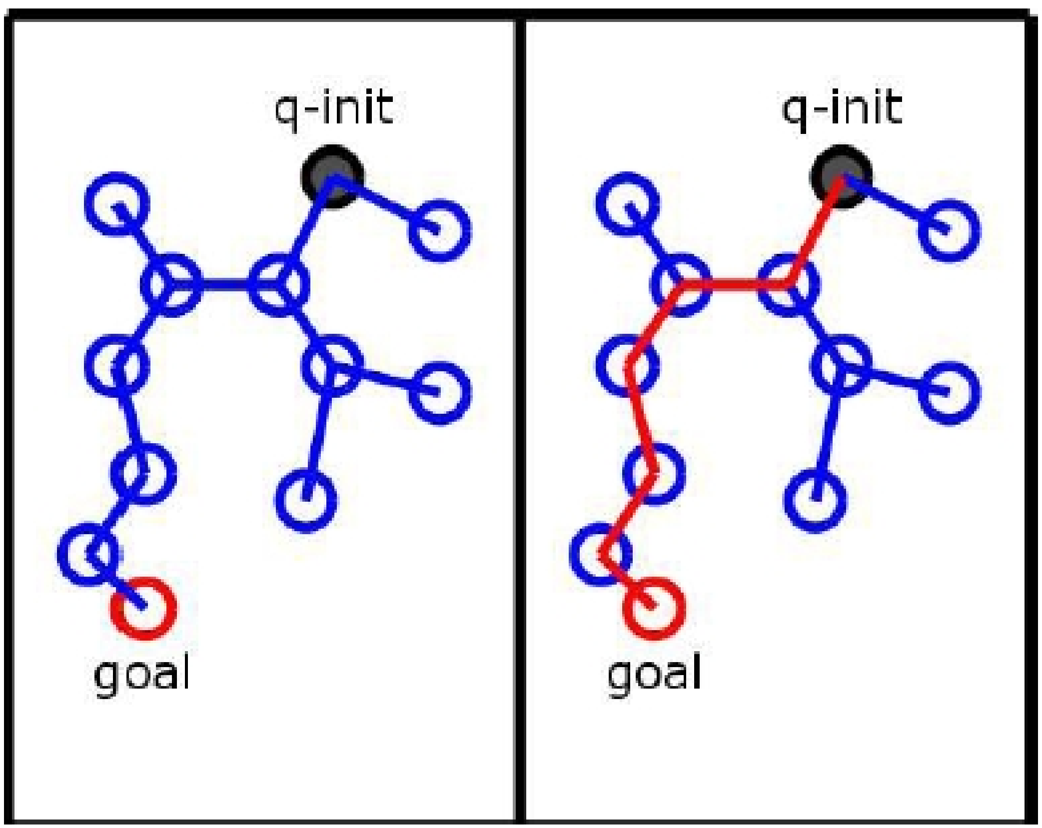
Ignore invalid extension



Record valid extension

# RRT As a Planner

- Once we reach the goal, follow the path back up the tree



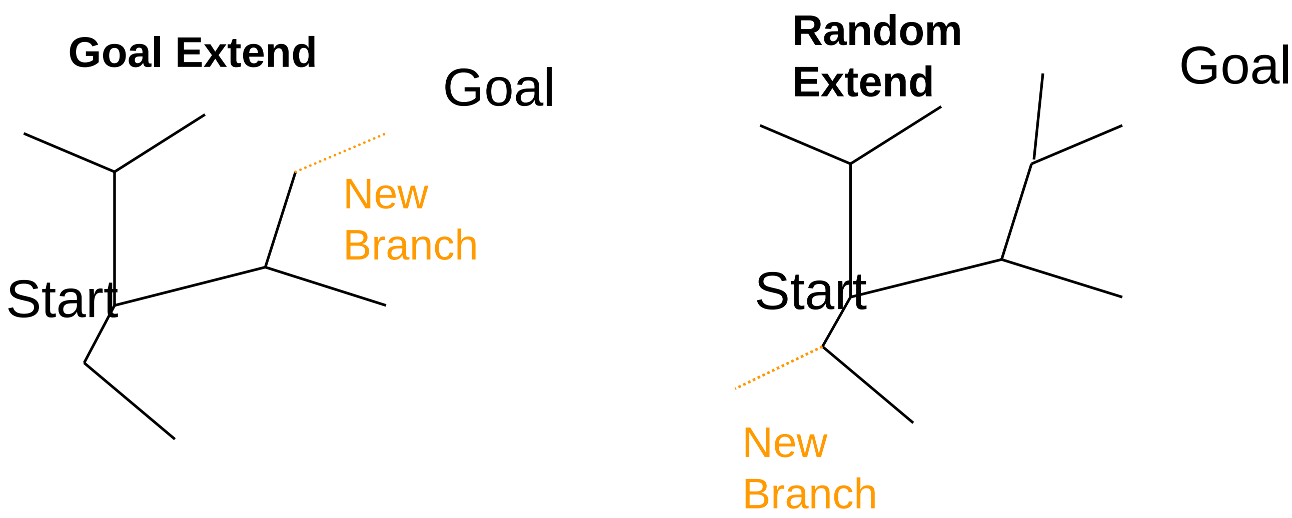
# RRT-GoalBias Algorithm

1) Start with initial state as root of tree
2) Pick a random target state
   - Goal configuration with probability p
   - Random configuration with probability 1-p
3) Find the closest node in the tree
4) Extend the closest node toward the target
5) Goto step 2

# RRT for Planning

Probability *p* : Extend closest node in tree towards goal

Probability *1-p* : Extend closest node towards a random point



# ERRT – RRT with Replanning

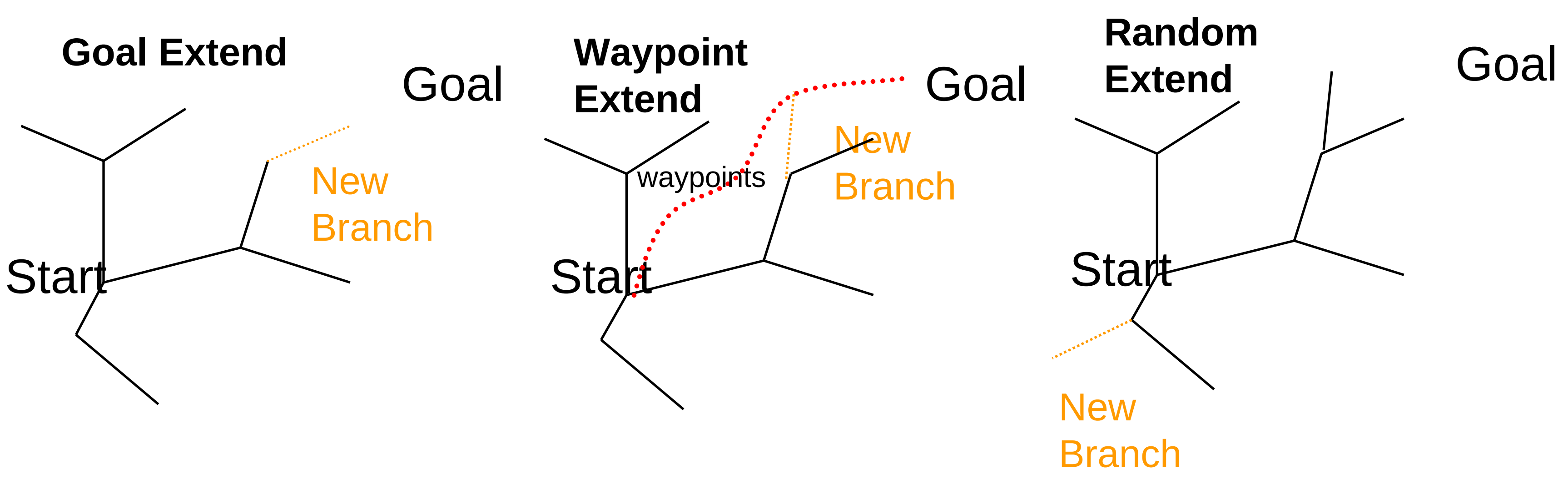
1) Start with initial state as root of tree
2) Pick a random target state
    - Goal configuration with probability p
    - Random item from waypoint cache with probability q
    - Random configuration with probability 1-q-p
3) Find the closest node in the tree
4) Extend the closest node toward the target
5) Goto step 2

# ERRT: Replanning with Advice

Probability *p* : Extend closest node in tree towards goal

Probability *r* : Extend closest node in tree towards random cache point

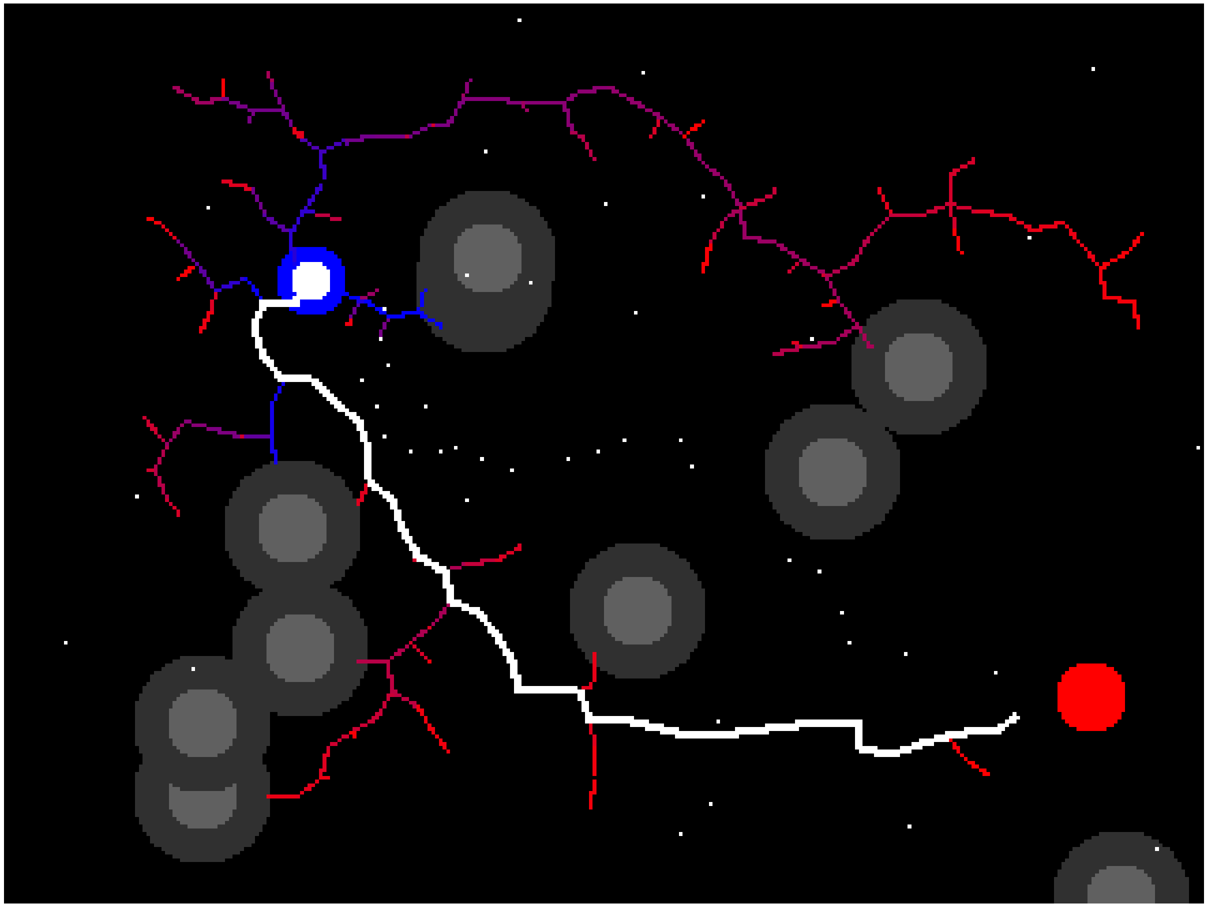
Probability *1-p-r* : Extend closest node towards a random point



# Discussion

- Planning with RRT
  - High p – few known obstacles
  - Low p – many known obstacles
- Replanning with ERRT
  - High q – small dynamics (no state change)
  - Low q – high dynamics (lots of state change)
  - ERRT – bias to use previous plan; but could be any other bias
- RRT and ERRT – probabilistic convergence

# Path Planning and Replanning



# Path Planning Conclusion

- Problem solving
- Path planning – states, actions, heuristics
- Probabilistic path planning
- Replanning