CMRoboBits:
Robot Behaviors, Planning

Manuela Veloso
Scott Lenser

15-491
November 17, 2003

Planning

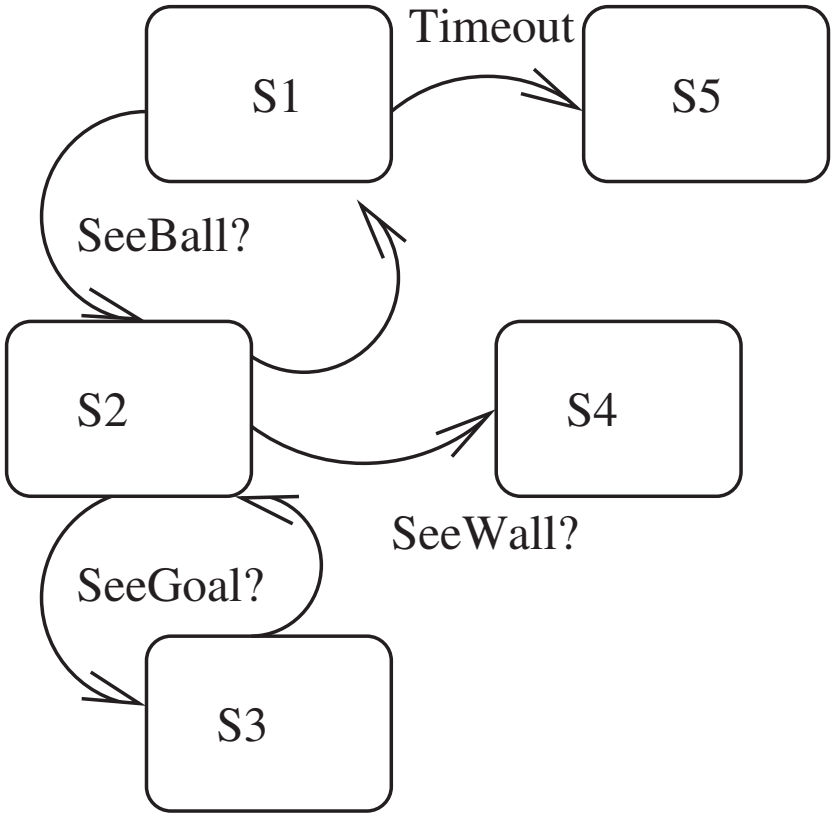
- Planning: selecting one sequence of actions (operators) that transform (apply to) an initial state to a final state where the goal statement is true.
- Different "levels" of planning: "low-level" motion planning, path planning, "high-level" planning, life planning!
- Action models - preconds, adds, dels, deterministic, probabilistic effects.
- State representation - symbolic, discrete, deterministic
- Goal statement - symbolic, conjunctive, state-based
- Algorithms
  - Backward chaining, forward chaining, search, soundness, completeness
  - Conditional, probabilistic planning
  - Interleaving planning and execution

Reactive Planning: Finite-State Machines

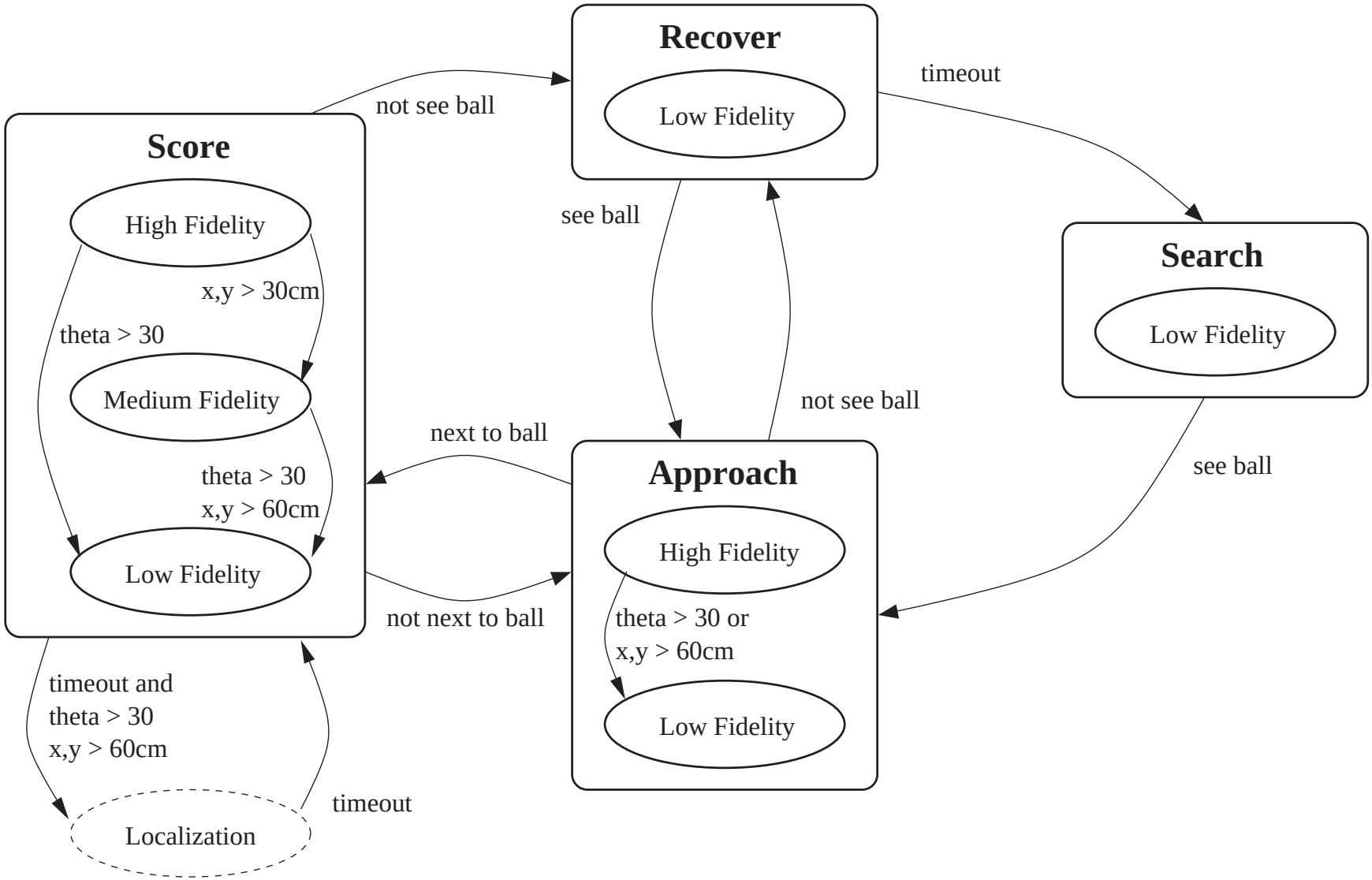
Color thresholding,
object recognition

CMVision
State Assessment
World Modeling

Purposeful Perception



Multi-Fidelity Behaviors



Multi-Fidelity Behaviors

Approach:

Low:

```
Run straight towards the ball.
```

High:

```
Skew approach to ball to get behind it,
when closer to its goal position.
```

Score:

Low:

```
Until the robot sees the goal,
  Walk sideways around the ball.
Walk forward pushing the ball.
```

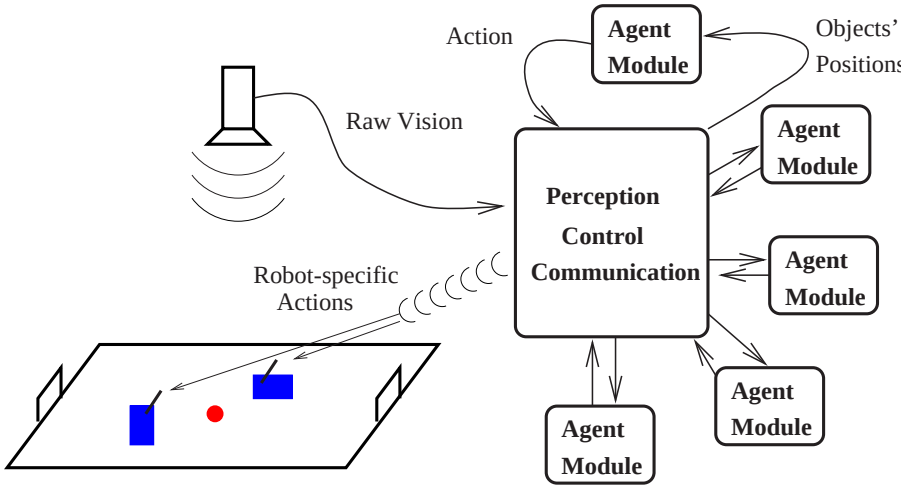
Medium-High:

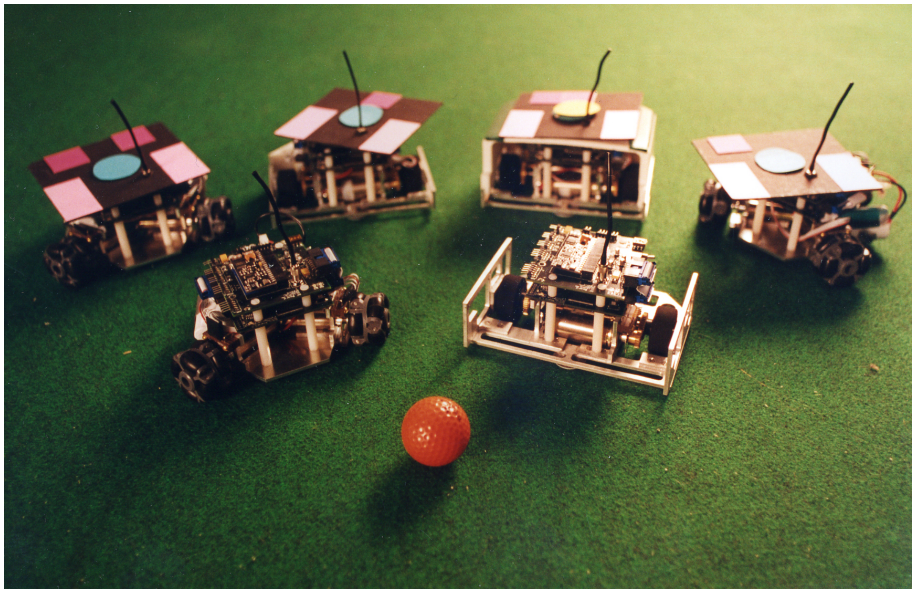
```
Circle ball using shortest distance
If facing goal, push ball forwards.
```

Multi-Fidelity Behaviors

- Explicit reasoning about sensor uncertainty
- Multiple finite state machines
- Multiple instantiations of a "behavior" as a function of robot's confidence on its sensing
- Explicit reasoning about reduce uncertainty on demand

CM-Dragons Small Soccer Robots

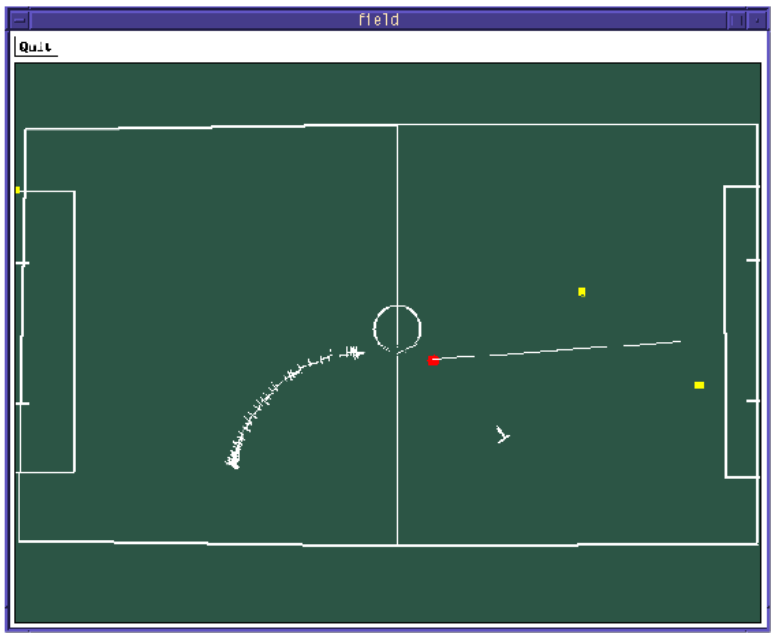




- Perception allowed through global vision
- Off board sensing and reasoning
- Complete cycle is autonomous

Planning

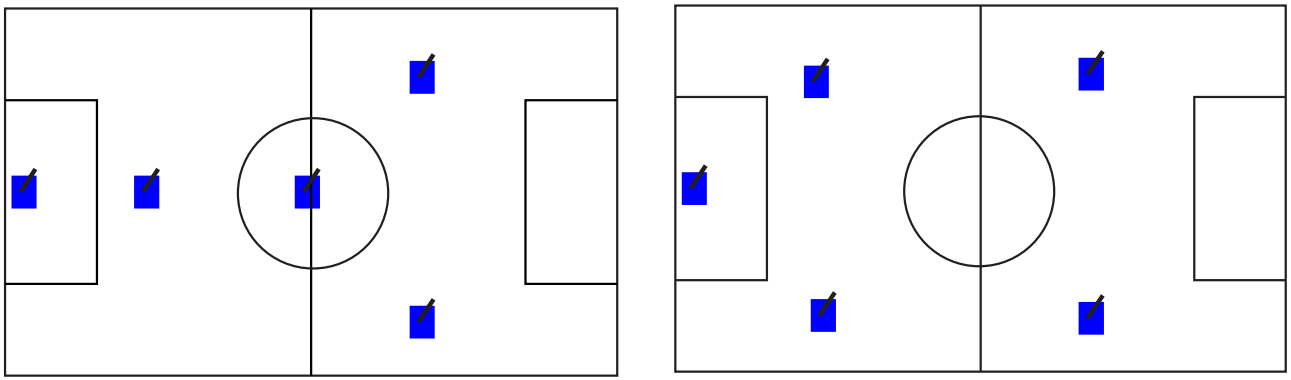
- State is detected 30-60 frames per second.
- Actions need to be selected at the same rate.
- Prediction for changes
  is valid for a few seconds.



- "High-level" planning
  - follow contingency plans.
- "Low-level" planning
  - go to point, navigate.

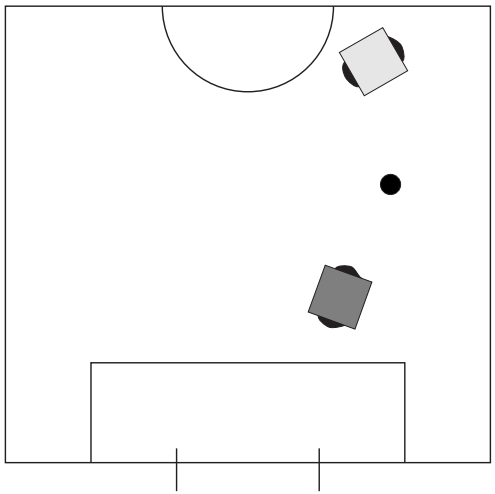
Team Organization

- Roles
  - Goalie, defender, attacker
  - Different formations

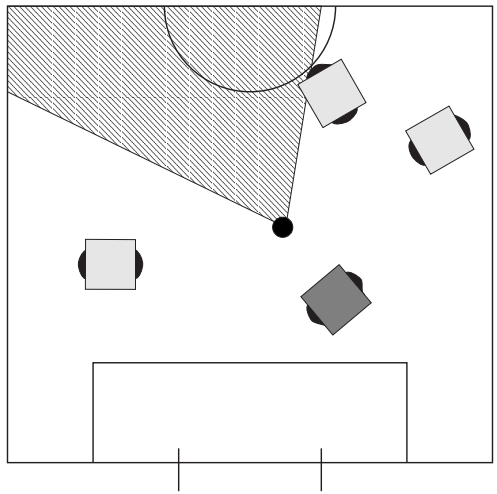


Reactive Defending Behaviors

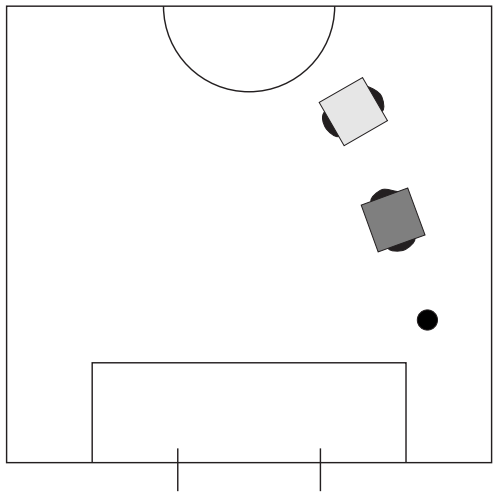
- BLOCK : Move between the ball and the goal.
- CLEAR : If open angle, shoot forward.
- ANNOY : Move between opponent and ball.



BLOCK
(Second goalie)

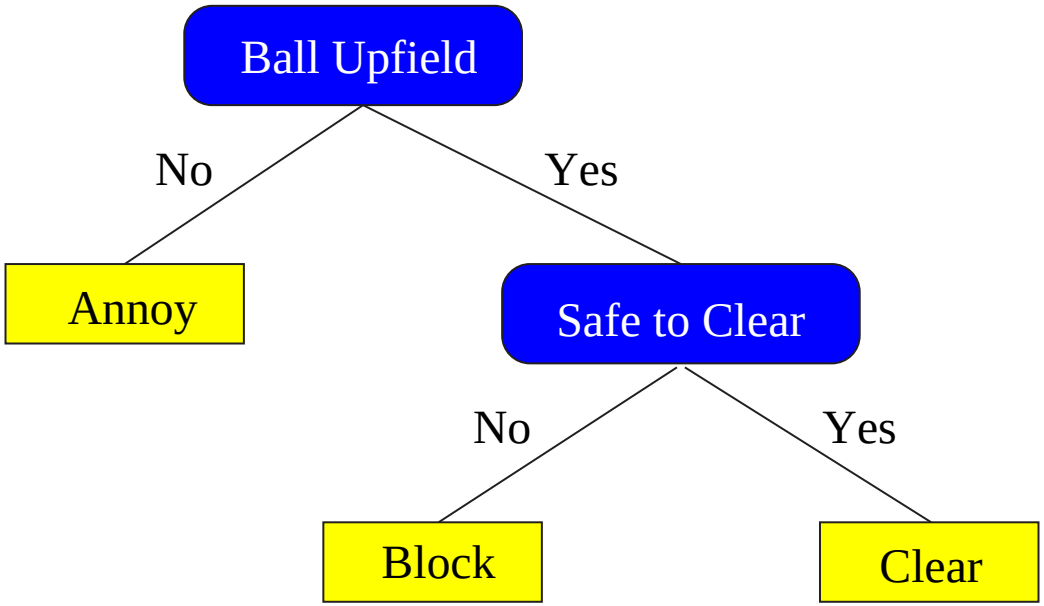


CLEAR
(Help attackers)



ANNOY
(Trouble opponent)

Continuous Contingency Plan



Planning with (Short) Lookahead

Attacking actions: Shoot, Pass

- Continuous reactive evaluation:
  - Maximize the probability of scoring discounted by the expected time to achieve the action.
  - Planning with a **two-step** lookahed
  - N robots, N possible actions, N^2 values

Example of Action Selection with Two Attackers

Attacker	Action	Probability of Success		Time(s)	Value
		Pass	Shoot		
1	Shoot	–	60%	2.0	0.30
1**	Pass to 2	60%	90%	1.0	0.54
2	Shoot	–	80%	1.5	0.53
2	Pass to 1	50%	40%	0.8	0.25

Planning as Ojective Satisfaction

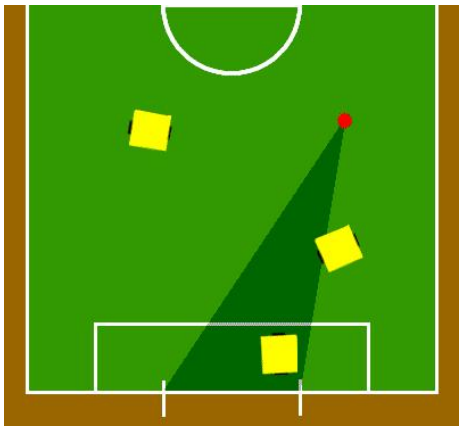
Move to a "good open" position

Repulsion	Maximize distance to opponents O_i
	Maximize distance to teammates T_i
Attraction	Minimize distance to ball $d(P, B)$
	Minimize distance to goal $d(P, G)$
Constraints	rules, legal positioning, etc.

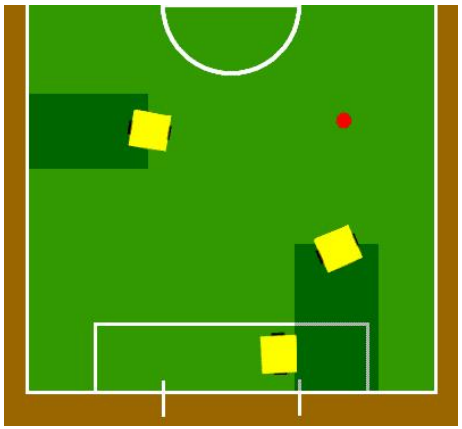
Find P, such that:

$$\max \left(\begin{array}{c} \sum_{i=1}^{n} w_{O_i} dist(P, O_i) + \sum_{i=1}^{n} w_{T_i} dist(P, T_i) - \\ -w_B dist(P, B) - w_G dist(P, G) \end{array} \right)$$

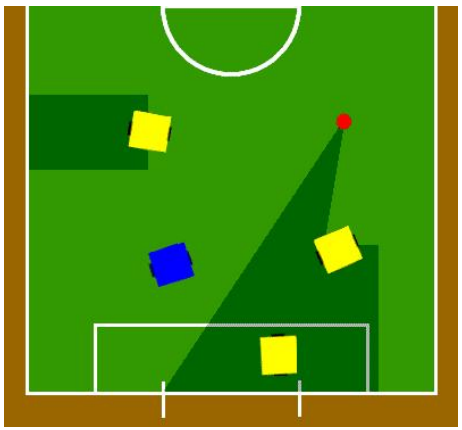
Objective Constrained Optimization



Do not block goal



Do not compromise passes

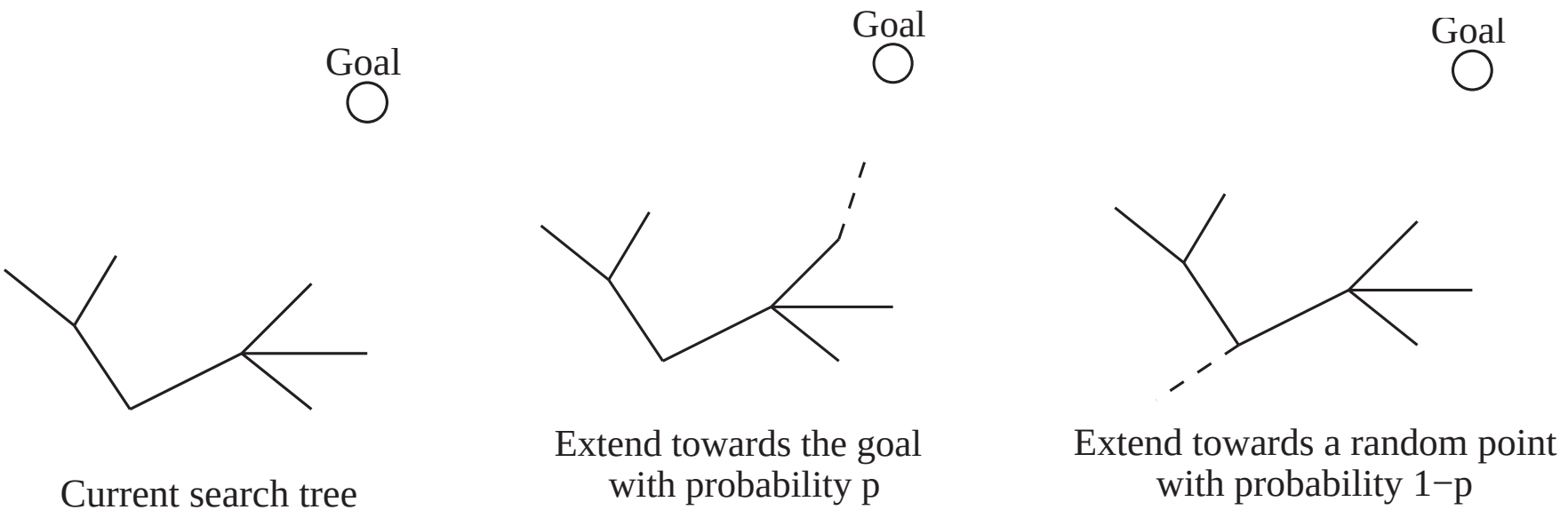


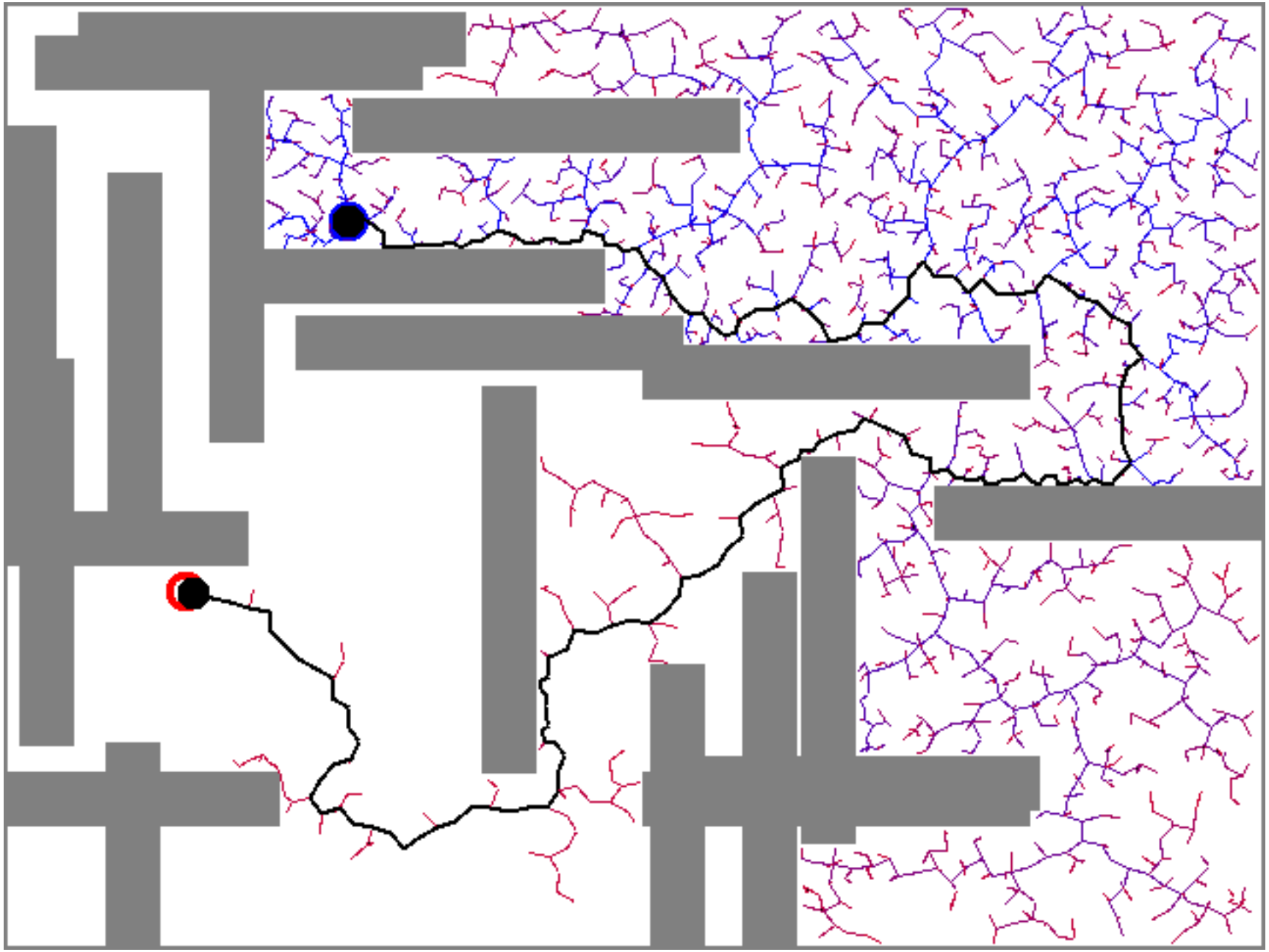
Optimize multi-objective function
under combined constraints

Low-level Planning - Navigation

- Real-time continuous path planning not fully solved
- RRT - Rapidly-exploring Random Trees [Lavelle 98]
- Basic RRT algorithm:

  goal-directed search : with probability p, extend closest node to goal
  random search : with probability $1 - p$, extend closest node to a random point





RRT Discussion

- Near optimal solutions, but not smooth
- Fast, but not fast enough for real robot navigation
- Impossible to expand complete path in real-time
- Need for replanning

Our contribution - extend RRT to robot navigation:

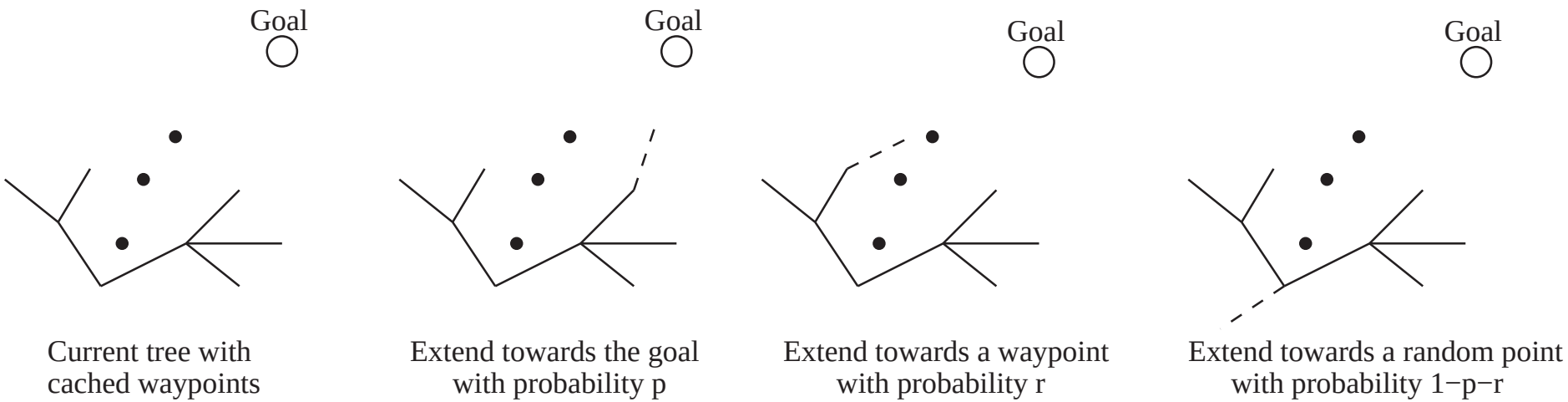
- **Post planning smoothing**
- **KD-trees for efficient data storage and access**
- **Replanning - cache waypoints!**

E-RRT - RRT for Robot Navigation

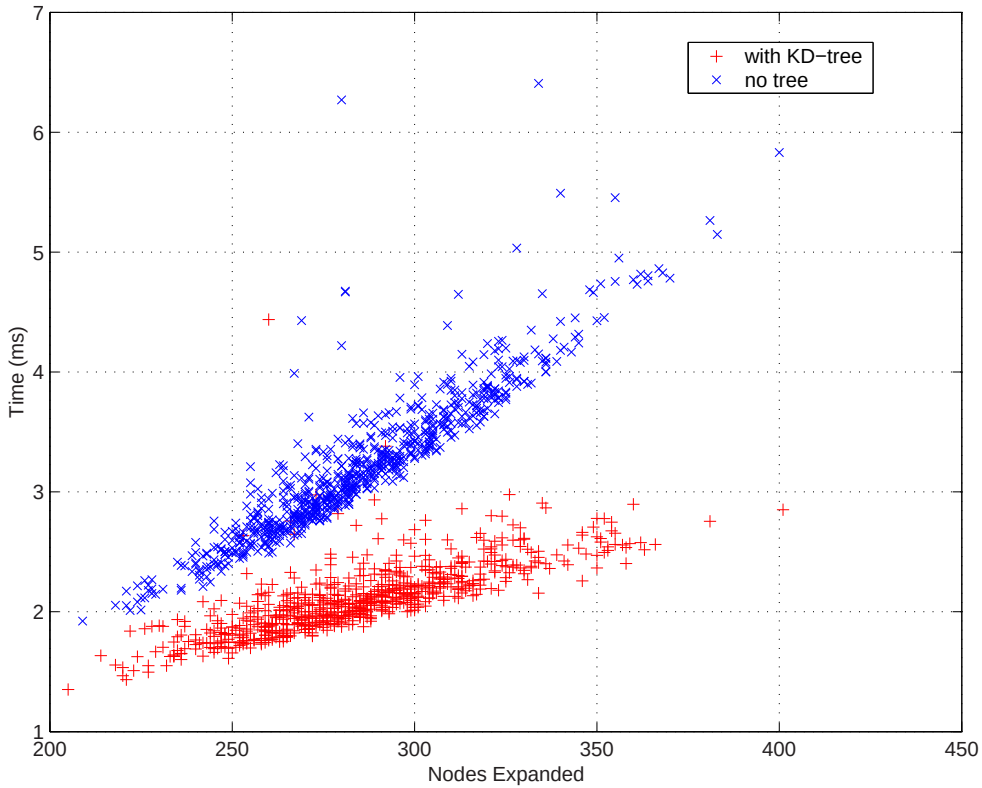
goal-directed : with probability p, extend closest node to goal

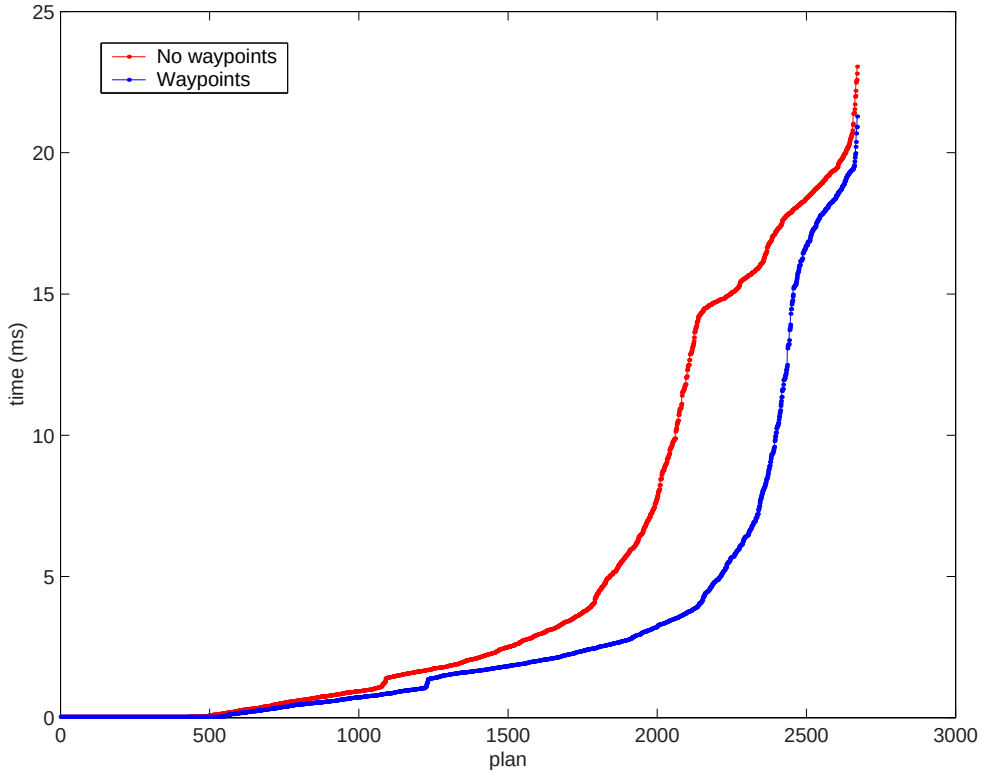
replan : with probability r, extend closest node to a random **cached** waypoint

random : with probability $1 - p - r$, extend closest node to a random point

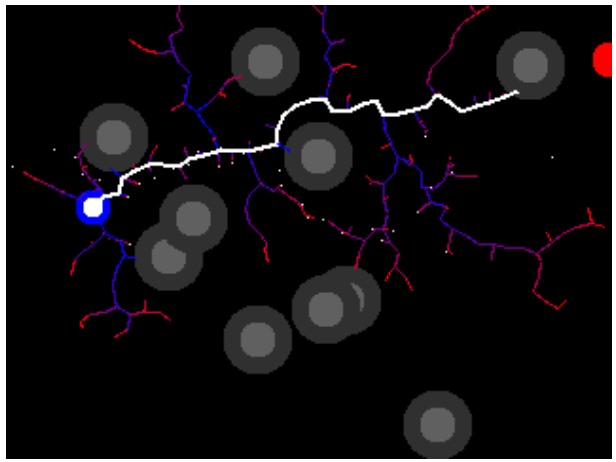


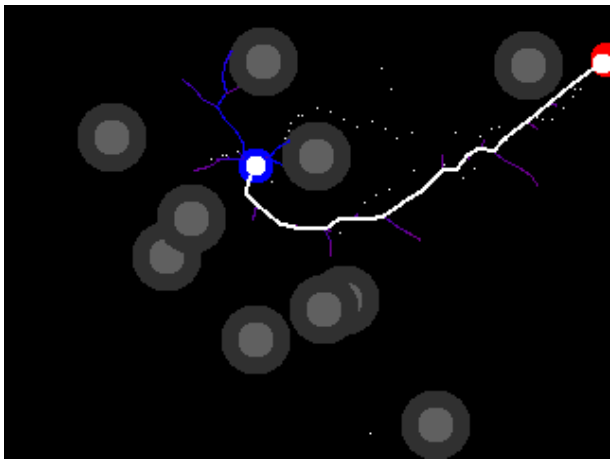
E-RRT Storage, Replanning Efficiency

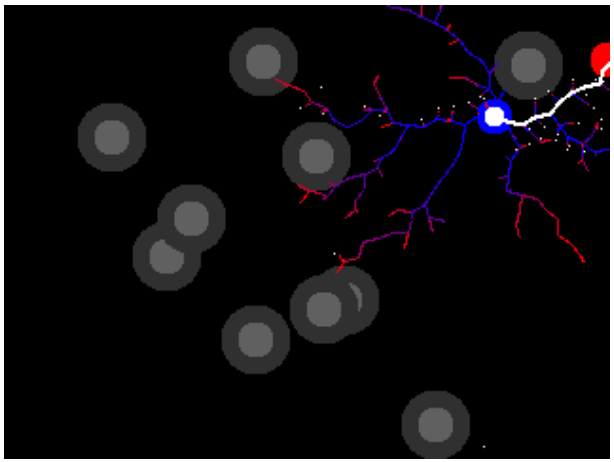




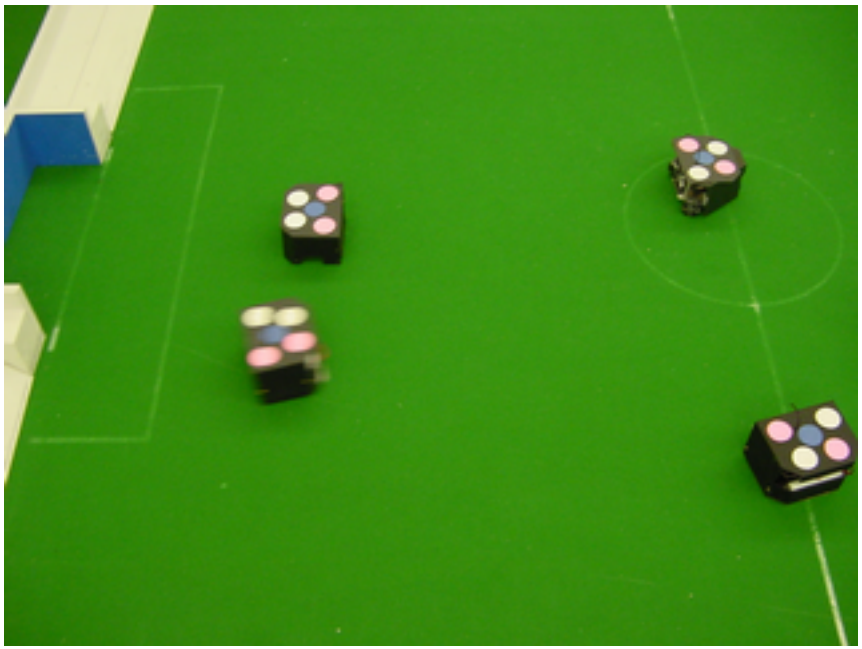
E-RRT - Robot Simulation







E-RRT - Multi-Robot Navigation



Summary

- Robot planning has a strong component of processing of perceptual information for state assessment.
- Finite-state machines provide plans/behaviors to robots.
- Uncertainty in the sensory information (state assessment) can be handled by multi-fidelity behaviors.
- Robot planning occurs at several "levels" of abstraction: low-level navigation/path planning, behaviors, multi-robot coordination.