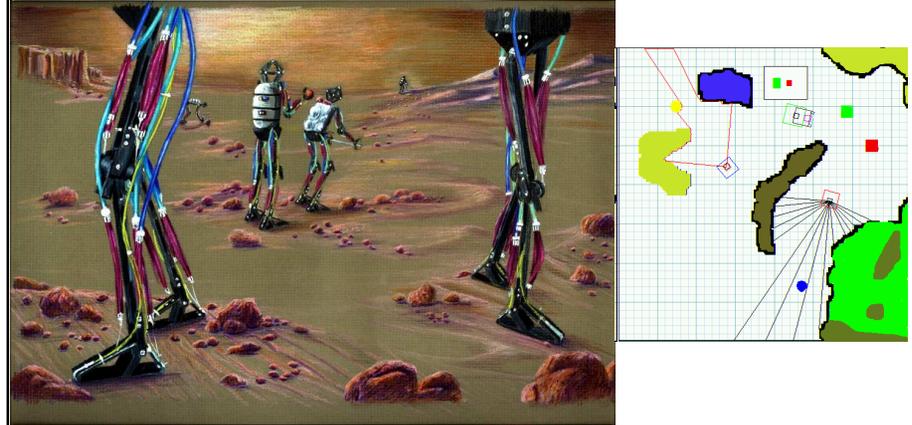


Team Coordination Strategies

PI: Dr. Hamid Berenji CO-I: David Vengerov, Jayesh Ametha (IIS/ARC)

Goal: Develop multi-agent teams that can coordinate and cooperate while performing a complex task. The work should allow teams of heterogeneous agents with differing expertise to work together to achieve common goals.

Objectives: Algorithms for teams of rovers to explore Mars or for the coordination between satellites either observing earth or deep space events. The need is for these resources to interact to achieve their goal since one of the agents by themselves may require additional support from platforms with different resources.



NASA Relevance:

- Enable science missions with multiple rovers to explore planetary surfaces. Work with Dr. Seraji's team at JPL on Mars missions prototypes. Provide algorithms for optimal power control algorithms for their wireless communications

Accomplishments to date:

- A convergent actor critic generalized reinforcement learning (ACFRL) algorithm has been developed, peer reviewed, and will appear in IEEE Transaction. Applied to wireless power control, substantially improves the networks performance Have published 6 papers and a book chapter on the theory.

Schedule:

- FY01: Develop a generalized reinforcement learning methodology for multi agent cooperation and coordination
- FY02: Demonstrate for teams of rovers to explore Mars
- FY03: Develop a prototype for a team of hardware and virtual agents for Mars exploration working, learning, and optimizing the team's performance. Demonstrate the prototype jointly with JPL

A Team of Heterogeneous Agents

- Each agent use a perception based rule set
- Agents can be modeled to have different capabilities and expertise
- Some can be sensory rich (many sensors, several rule preconditions) and/or knowledge rich (many rules)

Simulation Software

- The software applications used in this work, are 'Player' and 'Stage' that were developed jointly at the USC Robotics Research Lab and HRL Labs
- Player is a multi-threaded robot device server. It gives simple and complete control over the physical sensors and actuators on a mobile robot

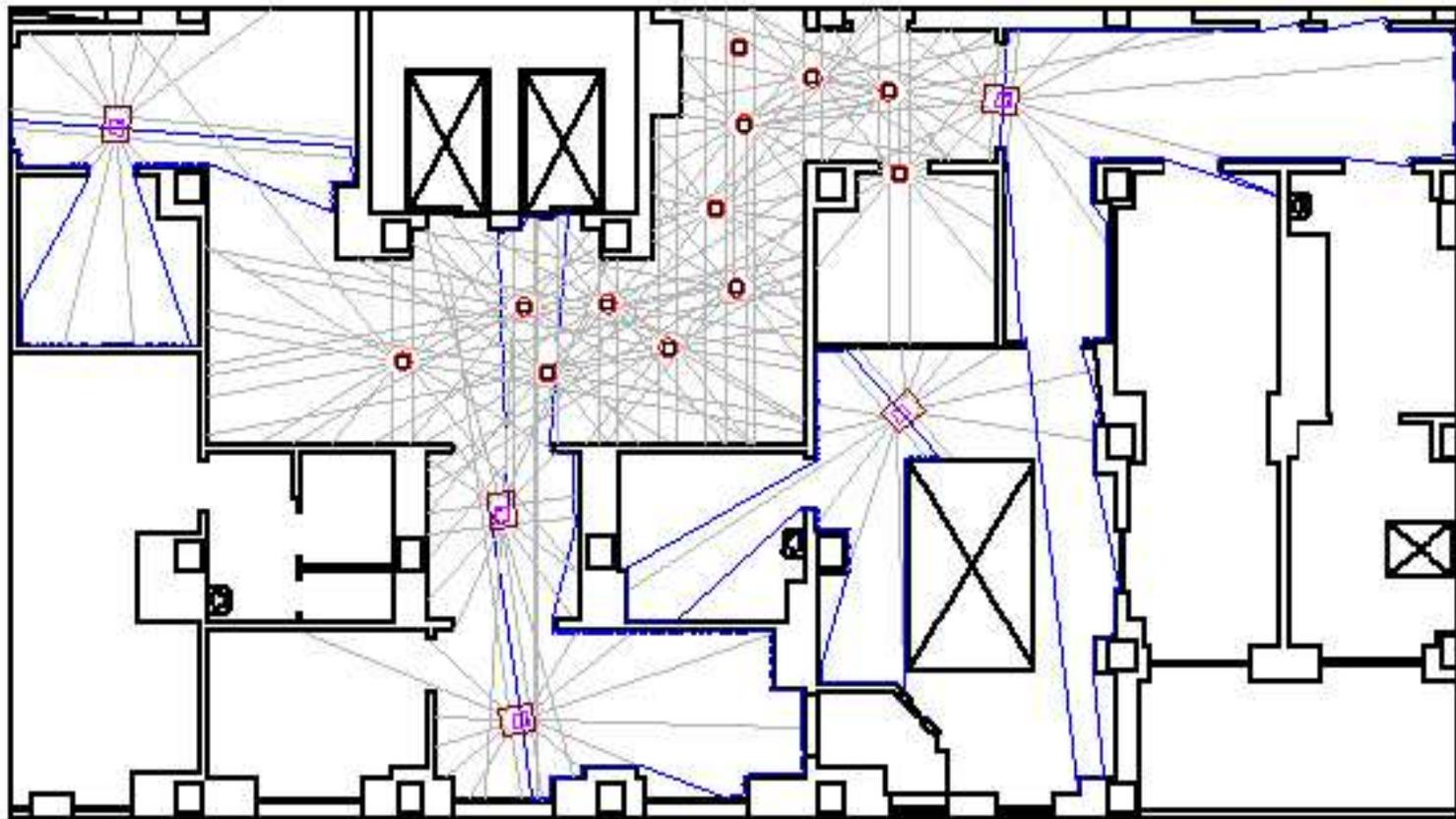
Player

- Player is also designed to support virtually any number of clients.
- Robots can "see" through each other's eyes.
- Any client can connect to and read sensor data from (and even write motor commands to) any instance of Player on any robot.

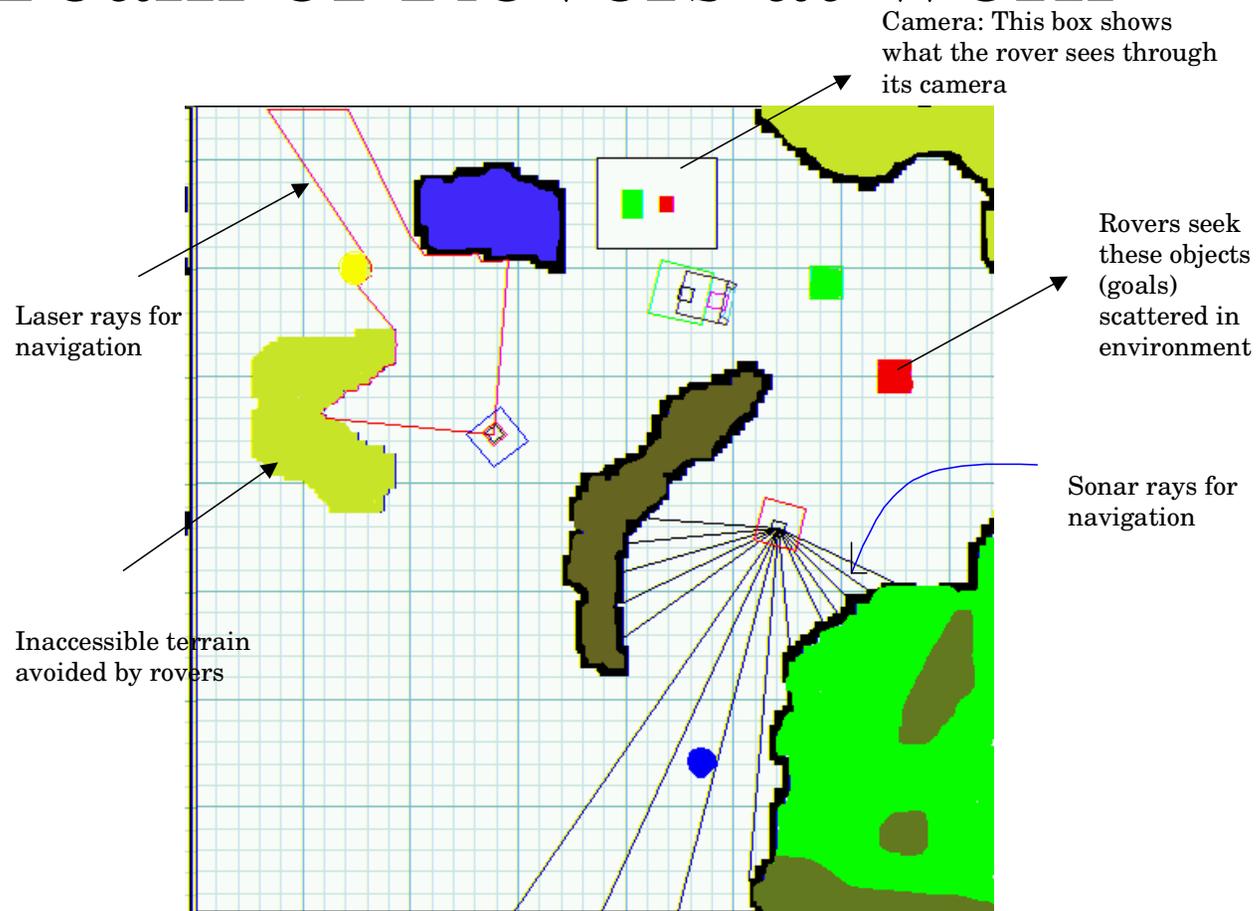
Player/Stage

- Our control program for the simulator can be used without any changes on the real robots.
- Stage simulates a population of mobile robots, sensors and objects in a two-dimensional bitmapped environment.

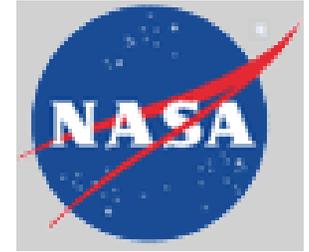
Floor navigation



A Team of Rovers at work



A team of heterogeneous autonomous rovers explores an unknown environment searching for certain valuable rocks (goals). As shown in figure, they use various sensors such as sonar, lasers, and camera among others.



USING FUZZY REINFORCEMENT LEARNING FOR POWER CONTROL IN WIRELESS TRANSMITTERS

David Vengerov
Hamid Berenji

Reinforcement Learning

- Decision policy specifies what action to take in each state of the environment
- Reinforcement learning -- learning an optimal decision policy based on reinforcements received from environment

A Brief History of Reinforcement Learning

- Actor-Critic algorithms appeared in early 1980s with no convergence proof
- Q-learning appeared in 1989 and suggested combining actor and critic into a single measure: Q-value
- Convergence results were obtained for Q-learning in finite MDPs with no function approximation

State Generalization

- In large state spaces, most states will be visited only once
- Need to generalize learning experience across similar states
- Function approximation for generalizing state values

Limitations of Q-learning With State Generalization

- Q-learning can diverge even for linear approximation architectures
- Requires solving a nonlinear programming problem at each time step when action space is continuous

Actor-Critic Algorithms

- Actor-critic (AC) algorithms can be used in continuous action spaces because actor can be parameterized
- Tsitsiklis and Konda (1999) presented a practical convergent AC algorithm
- Actor is a parameterized function that has to satisfy certain conditions

Actor-Critic Fuzzy Reinforcement Learning (ACFRL) algorithm

- Actor is represented by a fuzzy rulebase
- Convergence proven in Fuzz-IEEE 2001

Power Control for Wireless Transmitters

- Transmitter -- finite-buffer FIFO queue
- The transmission probability is a function increasing with power p_t and decreasing with channel interference i_t : $\text{Prob}(\text{success} \mid p_t, i_t) = 1 - e^{-\frac{p_t}{i_t}}$
- The transmission cost at time t is a function of transmitter's backlog b_t and the power used p_t :
 $C_t = \alpha p_t + b_t$.
- When a packet arrives to a full buffer, an overflow cost L is incurred.

Power Control for wireless transmitters

- Agent observes current interference i_t and backlog b_t and chooses a power level p_t
- Objective: minimize the average cost per time step.

Tradeoff to be learned

- Higher power incurs a higher immediate cost but also increases the probability of a successful transmission thereby reducing the future backlog.

Agent Structure

- An agent is a fuzzy rulebase, which specifies transmission power as a function of backlog(b) and interference(i):
 - If (b is SMALL) and (i is SMALL) then (power is p_1)
 - If (b is SMALL) and (i is MEDIUM) then (power is p_2)
 - If (b is SMALL) and (i is LARGE) then (power is p_3)
 - If (b is LARGE) and (i is SMALL) then (power is p_4)
 - If (b is LARGE) and (i is MEDIUM) then (power is p_5)
 - If (b is LARGE) and (i is LARGE) then (power is p_6)

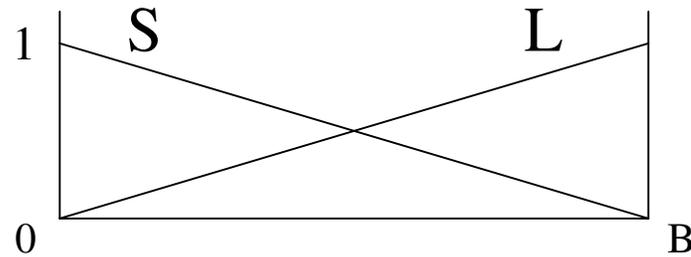
Motivation for the rulebase structure

- Bambos and Kandukuri (INFOCOM 2000) analytically derived a special-case power control policy:
 - Hump-shaped interference response resulting in a backoff behavior
 - The size of the hump grows with backlog

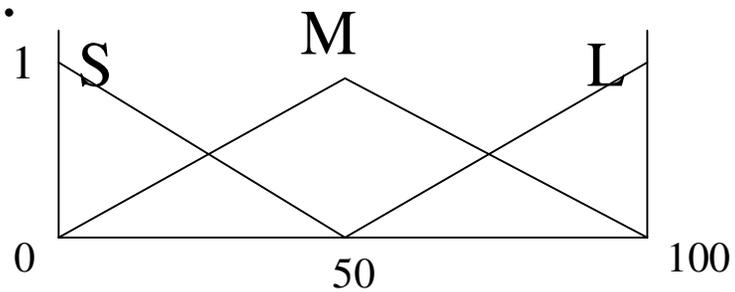
Labels

- Input labels:

- backlog:



- interference:



Simulation Procedure

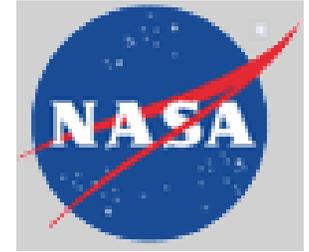
- Determine optimal constant power p^*
- Initialize p_1, \dots, p_6 to p^*
- Let ACFRL tune p_1, \dots, p_6

Problem Parameters

- Problem setup of Bambos and Kandukuri:
 - Poisson arrivals, uniform i.i.d. interference, finite buffer
- Simulated arrival rates 0.1 through 0.6, corresponding to low and high stress levels on the transmitter

Results

- ACFRL learns a hump-shaped interference response
- The size of the hump grows with backlog
- Corresponds to a special-case analytical study by Bambos and Kandukuri



Adaptive Coordination Among Fuzzy Reinforcement Learning Agents

David Vengerov

Hamid Berenji

Alexander Vengerov

Task distribution in multi-agent systems

- Traditional task distribution in multi-agent systems:
 - Centralized allocation
 - Allocation by auction (directly or through brokers)
 - Allocation by acquaintances
- Works well in static, known environments

Emergent allocation methods

- Interested in dynamic, a priori unknown environments
- Emergent allocation methods: signal-based rather than message-based.
- Agents learn the value of signals in the context of their local environments

Q-learning

- $Q(s, a)$ is the expected reward in state s after taking action a and following the optimal policy thereafter:

$$\begin{aligned} Q(s, a) &= E\{R_t \mid s_t = s, a_t = a\} \\ &= E\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s, a_t = a\right\} \end{aligned}$$

- r_t : the reward received after taking action a in state s_t ,
- γ : is the discounting factor.

Q-learning

- In discrete state and action spaces:

$$Q(s_t, a) \leftarrow Q(s_t, a) + \alpha_t (r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a)),$$

- α_t : is the learning rate at time t .
- Converges to optimal Q-values (Watkins, 1989) if each action is tried in each state infinitely many times,

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \quad \sum_{t=0}^{\infty} \alpha_t < \infty.$$

State Generalization

- In large state spaces, most states will be visited only once
- Need to generalize learning experience across similar states
- Function approximation for generalizing state values

Q-learning with state generalization

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_t - \frac{1}{2} \alpha_t \nabla_{\boldsymbol{\theta}_t} [r_{t+1} + \gamma \max_a Q(s_{t+1}, a, \boldsymbol{\theta}_t) - Q(s_t, a_t, \boldsymbol{\theta}_t)]^2.$$

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_t + \alpha_t \nabla_{\boldsymbol{\theta}_t} Q(s_t, a_t, \boldsymbol{\theta}_t) [r_{t+1} + \gamma \max_a Q(s_{t+1}, a, \boldsymbol{\theta}_t) - Q(s_t, a_t, \boldsymbol{\theta}_t)].$$

- $Q(s, a, \boldsymbol{\theta})$ approximates $Q(s, a)$
- $\boldsymbol{\theta}$ is the set of all parameters arranged in a single vector.

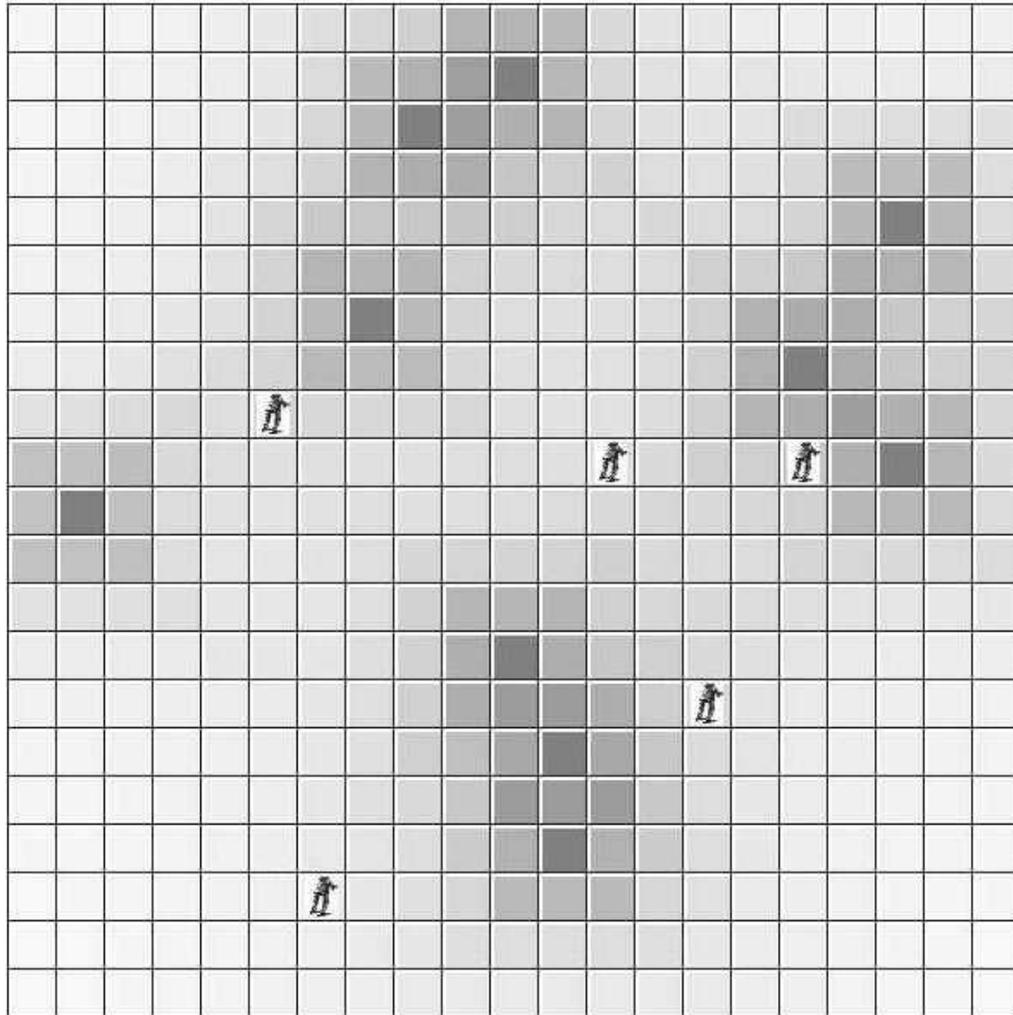
Distributed Dynamic Web Caching

- Servers distributed throughout the Internet
- Replicate content for faster access
- Main focus so far: directing requests to the “best” server
- Important issue: dynamically moving relevant content to servers located in “hot spots”

Agent-based View

- Agents represent content blocks
- Need to allocate themselves in proportion to the demand in each area
- Natural tradeoff for an agent:
 - moving to the highest demand area
 - ensuring adequate coverage of the whole area by the team

Tileworld Simulation



Tileworld Description

- Demand sources appear and disappear randomly
- Location-based similarity of interests
- Potential field model: demand source i contributes demand potential to location j :

$$P_{ij} = \frac{V_j}{1 + d_{ij}^2}$$

- Total potential at each location:

$$P_j = \sum_i P_{ij}$$

Tileworld Description

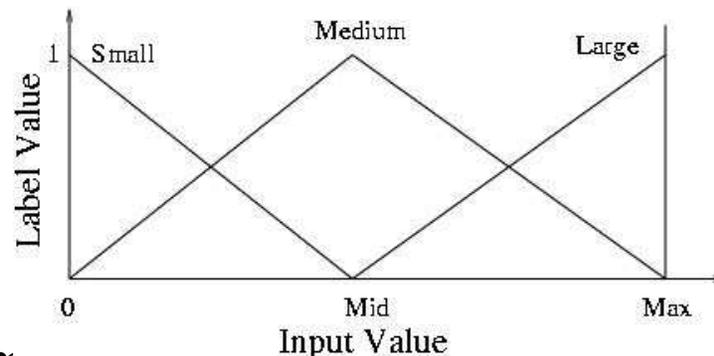
- Agent at location i extracts reward from source j equal to P_{ij}
- The value of each demand source decreases at each time step by the total reward extracted by all agents from this source
- Agent's goal: maximize average reward per time step

Agent Coordination

- Information about the team is presented to each agent in the form of “agent potential”
- Just like demand potential with agents being the sources

Decision Making

- Agents evaluate 8 adjacent locations
- Sample rule k: IF (demand potential at L_i is LARGE) and (agent potential at L_i is SMALL) then (Q-value of moving to L_i is Q_k^i)



- Final value of moving to location L_i .

$$Q^i = \sum_k \mu_k^i Q_k^i$$

Experimental Setup

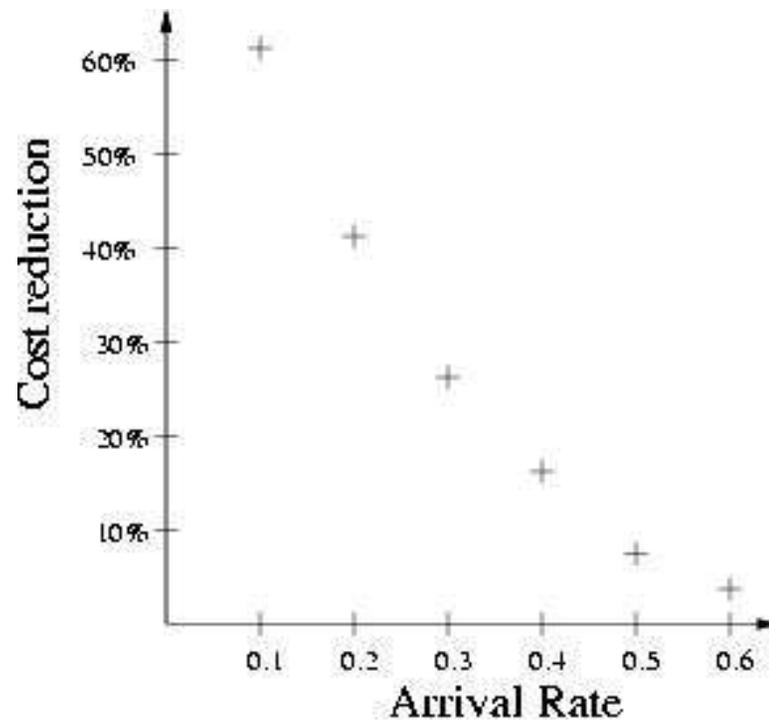
- 20-by-20 tileworld with 10 demand sources and 5 agents
- Agents are trained using Generalized Q-learning for 1000 time steps and then tested for 100 time steps
- Sensory radius: 5 units of distance or unlimited

Results

- Agents learn rules that prefer higher demand potential and smaller agent potential
- Coordinating agents perform 50-100% better than random agents
- Independent agents perform *worse* than random agents because they crowd together

Results

Cost improvement of ACFRL over optimal constant power policy:



For high arrival rates there is less freedom of buffering the arriving packets and waiting for better future channel conditions

Conclusions

- Demonstrated how ACFRL can be applied to a challenging delayed reward problem
- ACFRL converges to a policy that significantly improves upon optimal constant policy
- ACFRL learns the same function of the input variables as predicted by analytical investigations for a special case

Conclusions

- Fuzzy rulebased agents can learn successfully in continuous state spaces
- A new method for adaptive coordination among fuzzy reinforcement learning agents
- Agents learn an efficient group behavior in a dynamic resource allocation problem

Conclusion

- Perception based reasoning is crucial to develop smarter machines
- A fusion between perception based decision making and learning from the environment offers great potentials
- More heterogeneous inexpensive robotic agents need to be developed and be locally trained