Multiagent Coordination in Roombas: From a Neural Network Perspective

Jimmy Lin
Barry Feigenbaum

Prof. Risto Miikkulainen

Department of Computer Science
The University of Texas At Austin

December 1, 2014

Table of Contents

1. Problem Motivation
2. Approaches and Architectures
3. Experimental Results
4. Discussion

Section 1

Problem Motivation

Structure of the Roomba Environment

- Objects:
  - Agent: dynamic dirt collectors (Black Disks)
  - Scrumb: static dirt to clean (Blue Dots)
  - Wall: static boundary of the world
  - Chair/Desk: static decorations as hidden obstacles

  Roomba pictures
- Objectives:
  - collect as many scrumbs as possible
  - make as few collision as possible

A Ideal Case

- The most ideal case is to set up a system with
  - Centrailization: full control over all agents
  - Global View: full observations over all scrumbs
- In this system, the crumb collection can be formulated as an integer programming problem.
- No experiential learning is needed in this case.

A Slightly Realisitc Case

- Relax the "Centralization" assumption and yield a decentralized system with
  - Autonomy: all agents should decide on its own
  - Global View: full observations over all scrumbs are still available for all agents
- We need
  - Local policy optimizers that approximates the globally optimal policies
  - Collision-free or collision-tolerant protocol between agents

A More Realisitc Case

- Relax the "Global View" assumption, but allow limited information sharing between agents. These yield a system with
  - Autonomy: all agents should decide on its own
  - Local View: only local observation of scrumbs are available for agents
  - Limited Sharing: a small amount of information is shared between agents
- It is appropriate to solve the problem using experiential learning: reinforcement learning or neuroevolution.

Section 2

Approaches and Architectures

Challenges and Difficulties

The difficulty lies in

- Communication: what information to share for the best performance?
- ..

Approaches

baseline: random agents, greedy agents
more: neuroevolution, Q Learning
fitness derived from given number of episodes (each episodes with fixed number of actions).

Section 3

Experimental Results

We compared the fitness derived from

Section 4

Discussion

Discussion Time

- Any questions or suggestions?
  sert pictures
- Or email us at
  - jimmylin at utexas dot edu
  - chevron8 at gmail dot com

Acknowledgement

Thanks.